CURSO PRÁTICO 21 DE PROGRAMAÇÃO DE COMPUTADORES

INFORMÁTICA

PROGRAMAÇÃO BASIC - PROGRAMAÇÃO DE JOGOS - CÓDIGO DE MÁQUINA

Cz$ 39,00

INPUT

Vol. 2 Nº 21

NESTE NÚMERO

CÓDIGO DE MÁQUINA

ASSEMBLER PARA O MSX

Os montadores são programas tradutores que convertem os programas-fonte, escritos em Assembler, em programas-objeto, codificados em linguagem de máquina. Como calcular saltos e desvios. Use **POKE** para colocar os códigos na memória do computador 401

CÓDIGO DE MÁQUINA

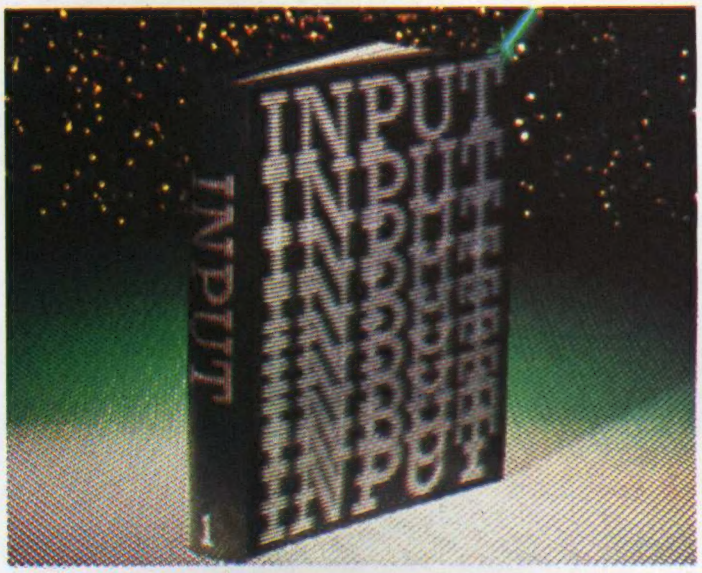
GRÁFICOS INSTANTÂNEOS

Como transformar desenhos em fileiras de números binários. Conversões de binário para decimal e para hexadecimal. Veja como é fácil animar jogos com figuras simples mesmo sem entender nada de sistema binário 406

APLICAÇÕES

UM ASSISTENTE DE ARTE

Desenhe com a ajuda do computador, executando o programa apresentado nesta lição. Você pode não chegar a ser um Picasso ou um Leonardo da Vinci, mas desenhará com desenvoltura auxiliado pelo seu micro doméstico 414



PLANO DA OBRA

"INPUT" é uma obra editada em fascículos semanais, e cada conjunto de 15 fascículos compõe um volume. A capa para encadernação de cada volume estará à venda oportunamente.

COMPLETE SUA COLEÇÃO

Exemplares atrasados, até seis meses após o encerramento da coleção, poderão ser comprados, a preços atualizados, da seguinte forma: **1. Pessoalmente** — por meio de seu jornaleiro ou dirigindo-se ao distribuidor local, cujo endereço poderá ser facilmente conseguido junto a qualquer jornaleiro de sua cidade. Em São Paulo os endereços são: Rua Brigadeiro Tobias, 773 (Centro); Av. Industrial, 117 (Santo André); e, no Rio de Janeiro: Rua da Passagem, 93 (Botafogo). **2. Por carta** — Poderão ser solicitados exemplares atrasados também por carta, que deve ser enviada para DINAP — Distribuidor Nacional de Publicações — Números Atrasados — Estrada Velha de Osasco, 132 (Jardim Tereza) — CEP 06000 — Osasco — São Paulo. **3. Por telex** — Utilize o nº (011) 33670 ABSA. Em **Portugal**, os pedidos devem ser feitos à Distribuidora Jardim de Publicações Ltd. — Qta. Pau Varais, Azinhaga de Fetais — 2685, Camarate — Lisboa; Tel. 257-2542 — Apartado 57 — Telex 43 069 JARLIS P.
Não envie pagamento antecipado. O atendimento será feito pelo reembolso postal e o pagamento, incluindo as despesas postais, deverá ser efetuado ao se retirar a encomenda na Agência do Correio. **Atenção**: Após seis meses do encerramento da coleção, os pedidos serão atendidos, dependendo da disponibilidade de estoque. **Obs.:** Quando pedir livros, mencione sempre o título e/ou o autor da obra, além do número da edição.

COLABORE CONOSCO

Encaminhe seus comentários, críticas, sugestões ou reclamações ao Serviço de Atendimento ao Leitor — Caixa Postal 9442, São Paulo — SP.

NOVA CULTURAL

Editor
VICTOR CIVITA

REDAÇÃO

Diretor Editorial: Carmo Chagas

Editores Executivos: Antonio José Filho, Berta Sztark Amar

Editor Chefe: Paulo de Almeida

Editor de Texto: Cláudio A. V. Cavalcanti

Chefe de Arte: Carlos Luiz Batista

Assistentes de Arte: Dagmar Bastos Sampaio, Grace Alonso Arruda, Monica Lenardon Corradi

Secretária de Redação/ Coordenadora: Stefania Crema

Secretários de Redação: Marisa Soares de Andrade, Mauro de Queiroz

COLABORADORES

Consultor Editorial Responsável: Dr. Renato M. E. Sabbatini (Diretor do Núcleo de Informática Biomédica da Universidade Estadual de Campinas)

Execução Editorial: DATAQUEST Assessoria em Informática Ltda., Campinas, SP

Tradução, adaptação, programação e redação: Abílio Pedro Neto, Aluísio J. Dornellas de Barros, Marcelo R. Pires Therezo, Marcos Huascar Velasco, Raul Neder Porrelli, Ricardo J. P. de Aquino Pereira

Coordenação Geral: Rejane Felizatti Sabbatini

Editora de Texto: Ana Lúcia B. de Lucena

COMERCIAL

Diretor Comercial: Roberto Martins Silveira

Gerente Comercial: Joaquim Celestino da Silva

Gerente de Circulação: Denise Maria Mozol

© Marshall Cavendish Limited 1984/85.
© Editora Nova Cultural Ltda., São Paulo, Brasil, 1986, 2ª edição, 1987.

Edição organizada pela Editora Nova Cultural Ltda.
Av. Brigadeiro Faria Lima, 2000 - 3º andar
CEP 01452 - São Paulo - SP - Brasil
(Artigo 15 da Lei 5988, de 14/12/1973).
Esta obra foi composta pela AM Produções Gráficas Ltda. e impressa pela Companhia Lithographica Ypiranga

ASSEMBLER PARA O MSX

Os montadores são programas tradutores encarregados de converter os programas-fonte, escritos em Assembly, em programas-objeto, codificados em linguagem de máquina.

Como já foi dito em artigos destinados a outros computadores, montar um programa em linguagem de máquina, calculando os códigos a mão, pode ser bem cansativo. Ainda que se saiba de cor todos os códigos mnemônicos e seus equivalentes hexadecimais e se esteja familiarizado com os modos de endereçamento, o trabalho de tradução e transferência do programa para o computador é sempre tedioso e possível de erros.

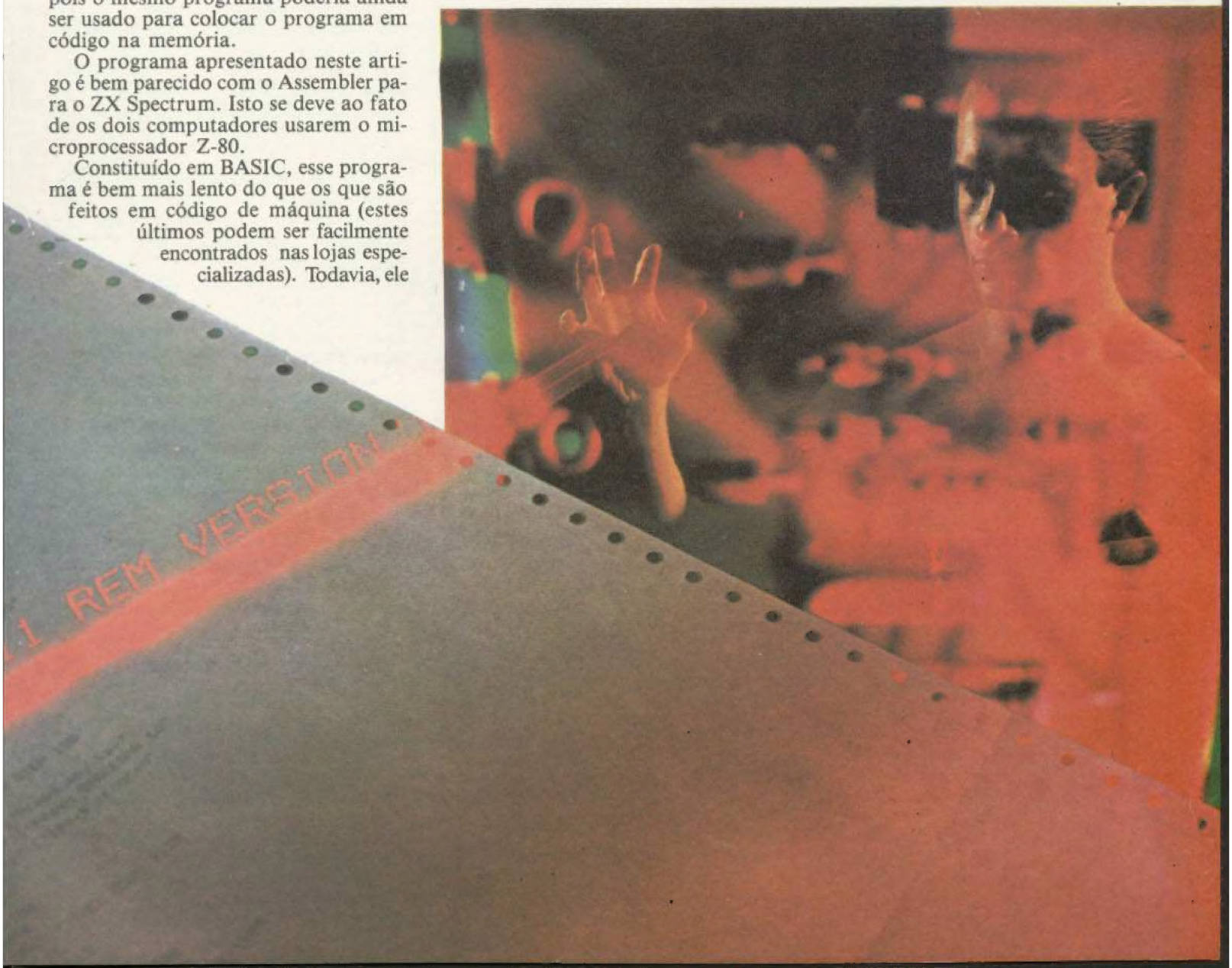
Já que computadores são muito bons nesse tipo de trabalho, por que não usá-los para fazer a tradução? De fato, só teríamos a ganhar se fizéssemos isso, pois o mesmo programa poderia ainda ser usado para colocar o programa em código na memória.

O programa apresentado neste artigo é bem parecido com o Assembler para o ZX Spectrum. Isto se deve ao fato de os dois computadores usarem o microprocessador Z-80.

Constituído em BASIC, esse programa é bem mais lento do que os que são feitos em código de máquina (estes últimos podem ser facilmente encontrados nas lojas especializadas). Todavia, ele funciona muito bem, embora devamos esperar algum tempo para que programas longos sejam montados na memória.

O ASSEMBLER

```
5000 CLS:KEYOFF:LOCATE8,10:PRIN
T"Um instante, por favor":CLEAR
500,&HDFFF:DIM K$(110),K(110),M
(110):H$="0123456789ABCDEF":B$=
"":G$="0123456789abcdef"
5010 DIMT$(100),R(100),Z$(100),
Z(100)
5020 B(1)=1:FORI=2TO9:B(I)=B(I-
1)+B(I-1):NEXT
5030 DIMR$(8,4):FORJ=1TO4:FORI=
1TO8:READR$(I,J):R$(I,J)=LEFT$(
R$(I,J)+"    ",4):NEXTI,J
5040 DATA 0,1,2,3,4,5,6,7,nz,z,
nc,c,po,pe,p,m,0,8,16,24,32,40,
48,56,hl,ix,iy,bc,de,hl,sp,"
5050 DIMS$(8,2),T(18),U$(18):FO
RJ=1TO2:FORI=1TO8/J:READS$(I,J)
:S$(I,J)=LEFT$(S$(I,J)+"    ",4
):NEXTI,J:FORJ=1TO18:READT(J),U
$(J):NEXTJ
5060 DATA b,c,d,e,h,l,(hl),a,bc
,de,hl,sp,235,de,8,af,227,(sp),
60742,0,60758,1,60766,2,233,(hl
```



O que acontecerá se houver um erro em meu programa-fonte?

Nosso Assembler é capaz de emitir mensagens de erro, pois a sua estrutura de programação permite reconhecer certas falhas no programa em Assembly. Alguns comandos, por exemplo, só trabalham com os registros **a** e **hl**. Se tentarmos usá-los com outros registros, o Assembler dirá: "Primeiro operando deve ser a ou hl".

Se um comando não for reconhecido devido a um erro de digitação, seremos informados: "Linha não reconhecida". A expressão "Deve haver dois operandos" significa que deixamos de digitar um número vital após o comando. "Operando incompatível" indica um uso inadequado do comando. "Primeiro operando deve ser sinalizador ou bit" significa que um operando inadequado foi empregado em um comando de desvio ou de atribuição de bits.

```
),56809,(ix),65001,(iy),10,(bc)
,26,(de),60767,r,2,(bc),18,(de)
,60751,r,249,sp,60743,i,60759,i
5070 DEFFNB(X,I)=INT(X/B(I+1))-
INT(X/B(I+2))*2
5080 DEFFNX(X,I)=X-B(I+2)*FNB(X
,I)+B(I+1)
5090 DEFFNJ(X,I)=INT(X)-B(I+1)*
INT(X/B(I+1))
5100 DEFFNE(I$,J$)=(I$=LEFT$(J$
,LEN(I$)))
5110 FORI=1TO110:READK$(I),K(I)
,M(I):IFNOTFNE("*",K$(I))THENNE
XTI
5120 DATA ld,10,10,ld,26,10,ld,
60767,10,ld,60759,10,ld,2,138,1
d,18,138,ld,60751,138,ld,60743,
138,ld,64,6,ld,50,202,ld,58,74,
ld,249,139,ld,34,195,ld,42,67,l
d,60779,197,ld,60771,199,ld,97,
165,ld,64,54
5130 DATA adc,136,50,adc,60746,
3,add,128,50,add,9,149,and,160,
48,or,176,48,xor,168,48,nop,0,0
,sub,144,48,sbc,152,50,sbc,6073
8,3,cp,184,48,jp,130,45,jp,233,
9,jp,56809,9,jp,65001,9,jp,131,
49,jr,96,45,jr,88,41
5140 DATA call,132,45,call,141,
41,ret,201,0,djnz,74,40,dec,11,
17,dec,5,16,inc,3,17,inc,4,16,p
ush,197,17,pop,193,17,di,243,0,
ei,251,0,halt,118,0,ex,235,139,
ex,8,15,ex,227,143,exx,217,0
5150 DATA rst,199,132,rts,192,5
,bit,52032,20,defb,-256,40,ccf,
63,0,scf,55,0,cpl,47,0,cpd,6084
1,0,cpdr,60857,0,cpi,60833,0,cp
ir,60849,0,daa,39,0,im,60742,8,
im,60758,8,im,60766,8
5160 DATA in,60736,130,in,149,4
2,ind,60848,0,indr,60810,0,ini,
60840,0,inir,60850,0,ldd,60840,
0,lddr,60856,0,ldi,60832,0,ldir
,60848,0,neg,60740,0,otdr,60859
,0,otir,60851,0,out,60737,2,out
,141,170,outd,60843,0,outi,6083
5,0
5170 DATA res,52096,20,reti,607
49,0,retn,60741,0,rl,51984,64,r
la,23,0,rlc,51968,16,rlca,7,0,r
ld,60783,0,rr,51992,64,rra,31,0
,rrc,51976,16,rrca,15,0,rrd,607
75,0
5180 DATA set,52160,20,sla,5200
0,16,sra,52008,16,srl,52024,16,
defw,-256,41
5190 DATA "*",0,0:II=I:K(110)=I
I
5200 CLS:PRINTTAB(15)"ASSEMBLER
":LOCATE10,5:PRINTTAB(10)"(c)
Ler na fita cassete":PRINT:PRIN
TTAB(10)"(g)  Gravar na fita":P
RINT:PRINTTAB(10)"(e)  Edição"
5210 PRINT:PRINTTAB(10)"(m)  Mo
ntar":PRINT:PRINTTAB(10)"(a)  A
pagar linha":PRINT:PRINTTAB(10)
"(l)  Listar":PRINT:PRINTTAB(10
)"(s)  Saída"
5220 A$=INKEY$:IFA$=""THEN5220
5230 JJ=INSTR("cgemals",A$)
5240 IFJJ=0THENPRINT"<"A$"> ???
?":FORJ=1TO500:NEXT:GOTO5200
5250 CLS:ONJJGOSUB10420,10450,1
0490,5290,10710,10760,10900
5260 PRINT:PRINT"Qualquer tecla
 para continuar"
5270 A$=INKEY$:IFA$=""THEN5270
5280 GOTO5200
5290 FORG=1TO100:R(G)=G-1:NEXTG
:Fh=100
5300 K0=0:K9=99:P0=0:VV=0
5310 K=K0:P=P0
5320 GOSUB8000
5330 GOSUB7000:O$=I$:IFLEFT$(O$
,1)="*" THENPRINTO$;:GOTO5320
5340 IFO$="end"THENPRINT:PRINT"
    FIM. Endereço final = ";P-1
5350 IFO$="end"THENP0=P:RETURN
5370 IFO$<>"org"THEN5400
5380 GOSUB7000:S=0:IFLEFT$(I$,1
)="*"THENS=P:I$=RIGHT$(I$,LEN(I
$)-1)
5390 P=VAL(I$)+S:PRINT"    org
";P;:GOTO5320
5400 IFP=0THENPRINT" (falta org
)":P=&HE000
5410 P$=O$+"!":FORI=1+18*ABS(O$
<>"ld")TO110:IFO$<=K$(I)ANDP$>K
$(I)THEN5500
5420 NEXTI:PRINTO$
5430 IFLEFT$(I$,1)="."THENI$=RI
GHT$(I$,LEN(I$)-1)
5440 GOSUB9000:GG=R(G)
5450 IFABS(GG)<=100THENS=SGN(Z(
GG)):B=INT(ABS(Z(GG))/65536!):R
=ABS(Z(GG))-B*65536!:Q=PEEK(R)+
256*PEEK(R+1):POKER,FNJ(P*S+Q,8
):PRINT" colocando ";FN J(P*S+Q
,8);" em ";R:IFBTHENPOKE(R+1),F
NJ((P*S+Q)/256,8):PRINT" coloca
ndo ";FNJ((P*S+Q)/256,8);" em "
;R+1
5460 IFABS(GG)<=100THENGH=R(GG)
:R(GG)=FH:FH=GG:GG=GH:GOTO5450
5470 IFI$=""THENR(G)=P+100:GOTO
5330
5480 PRINT" (Linha não reconhec
ida)"
5490 GOTO5420
5500 Z=0:R=0:E=0:PRINT"    ";O$
;
5510 OP=K(I):IFM(I)=0THEN6090
5520 GOSUB7000:A$=I$:PRINT" ";A
$;
5530 M=M(I):OP=K(I):B=FNB(M,0):
B7=B+2*FNB(M,7)+1:Z=0:IFFNJ(M,3
)<2THENC$=A$:GOTO5720
5540 FORJ=1TOLEN(A$):IFMID$(A$,
J,1)=","THEN5580
5550 NEXTJ:IFO$="rst"ORO$="rts"
THEN5580
5560 IFFNE(O$,K$(I+1))THENI=I+1
:GOTO5530
5570 PRINT" (são necessários do
is operandos)":GOTO5320
5580 B$=LEFT$(A$,J-1):C$=RIGHT$
(A$,LEN(A$)-J)
5590 IFFNB(M,2)THEN5650
5600 IFFNB(M,7)THEND$=C$:C$=B$:
B$=D$
5610 IFB$=MID$("ahl",B+1,B+1)TH
EN5720
5620 IFB$="(c)"AND(O$="in"ORO$=
"out")THEN5720
5630 IF(FNE(O$,K$(I+1)))AND(FNJ
(M(I+1),3)>=2)THENI=I+1:GOTO553
0
5640 PRINT" (primeiro operando
deve ser a ou hl)":GOTO5320
5650 IFFNB(M,1)THEN5690
5660 E$=LEFT$(B$+"    ",4):FORJ=
1TO8:IFE$=R$(J,B7)THENOP=OP+8*(
J-1)*ABS(B7<4)+16*(J-6)*ABS(B7=
4)*ABS(J>3):Z=(J-1)*ABS(B7=4)*A
BS(J<=3):GOTO5710
5670 NEXTJ:IFP$>K$(I+1)AND(FN J
(M(I+1),3)>=2)THENI=I+1:GOTO553
0
5680 PRINT" (primeiro operando
deve ser sinalizador ou bit)":G
OTO5320
5690 IFFNB(M,7)THEND$=C$:C$=B$:
B$=D$:GOTO5660
5700 X=8:GOSUB5750:IFETHEN5730
5710 IFC$=""THEN6090
5720 X=1+15*B+7*ABS(OP<=6ANDOP>
=40RB$="(c)"):B$=C$:GOSUB 5750:
IFABS((E=0)*NOTE)THEN6090
5730 IFE=20RP$>K$(I+1)ANDFNJ(M(
I+1),3)=FNJ(FNX(M,0),3)THENE=0:
I=I+1:GOTO5530
5740 GOTO5320
5750 R=0:IFFNB(M,4)ANDFNE(LEFT$
("(",ABS((B=0)*NOTB)),B$)THENZ2
=ABS(FNE("ix",RIGHT$(B$,LEN(B$)
+B-1)+" "))+2*ABS(FNE("iy",RIGH
T$(B$,LEN(B$)+B-1)+" ")):IFZ2TH
ENZ=Z2:E$=LEFT$(B$,LEN(B$)-ABS(
(B=0)*NOTB)):B$=MID$("(hl)",1+B
,4-2*B):F$="0"+RIGHT$(E$,LEN(E$
)+B-3)
5760 IFFNB(M,3)THEN5790
```

```
5770 E$=LEFT$(B$+"    ",4):FORJ="
1TO8/(B+1):IFE$=S$(J,B+1)THENOP
=OP+(J-1)*X:RETURN
5780 GOTO5810
5790 J2=9+9*ABS(O$="ld"):FORJ=J
2-8TOJ2:IFK(I)<>T(J)THEN5810
5800 IFFNE(B$,U$(J))THENRETURN
5810 NEXTJ:IFB$="af"THENIFFNE("
p",O$)THENOP=OP+48:RETURN
5820 IFFNB(M,6)ANDFNE("(",B$)TH
ENB$=MID$(B$,2,LEN(B$)-2):GOTO5
860
5830 IFFNB(M,5)THENOP=FNX(OP+6*
ABS((B=0)*NOT B),6):GOTO5860
5840 IFP$>K$(I+1)THENE=2:RETURN
5850 PRINT" (operando incompatí
vel)":E=1:RETURN
5860 R=65536!
5870 S=1
5880 IFB$=""THEN6080
5890 X$=LEFT$(B$,1):D$=RIGHT$(B
$,LEN(B$)-1):IFX$="*"THENR=R+P*
S:B$=D$:GOTO5870
5900 IFX$="+"THENB$=D$:GOTO5880
5910 IFX$="-"THENB$=D$:S=-S:GOT
O5880
5920 IFX$="'"THENR=R+ASC(D$)*S:
B$=RIGHT$(D$,LEN(D$)-1):GOTO587
0
5930 Q=0:IFX$<>"%"ORD$<"0"ORD$>
="2"THEN5960
5940 IFD$>="0"ANDD$<"2"THENQ=Q*
2+ASC(D$)-48:D$=RIGHT$(D$,LEN(D
$)-1):GOTO5940
5950 R=R+Q*S:B$=D$:GOTO5870
5960 IFX$<>"$"ORD$<"0"ORD$>="9"
THEN6000
5970 X$=CHR$(ASC(D$)):FORG=0TO1
5:IFX$<>MID$(H$,G+1,1)ANDX$<>MI
D$(G$,G+1,1)THEN5990
5980 Q=Q*16+G:D$=RIGHT$(D$,LEN(
D$)-1):GOTO5970
5990 NEXTG:R=R+Q*S:B$=D$:GOTO58
70
6000 IFX$<"a"ORX$>"z"THEN6040
6010 I$=B$:GOSUB9000:IFI$<>""TH
EN6010:GOSUB9400
6020 IFR(G)<>23000ANDABS(R(G))>
100THENR=R+(R(G)-100)*S:B$=I$:G
OTO5870
6030 IFR(G)=23000ORABS(R(G))<=1
00THENGH=R(FH):R(FH)=R(G):R(G)=
FH:FH=GH:Z(R(G))=(P+SGN(OP)+ABS
(ABS(OP)>255)+2*ABS(Z>0)+65536!
*ABS(((ABS(BORFNB(M,6))<>0)*1)A
ND(O$<>"jr")))*S:B$=I$:GOTO5870
6040 IFX$<"0"ORX$>"9"THENR=0:GO
TO6070
6050 IFB$>="0"ANDB$<":"THENQ=Q*
10+ASC(B$)-48:B$=RIGHT$(B$,LEN(
B$)-1):GOTO6050
6060 R=R+S*Q:GOTO5870
6070 PRINT" (endereçamento invá
lido)"
6080 R=R-(P+2)*ABS(O$="djnz"ORO
$="jr"):RETURN
6090 PRINTTAB(16);:BY=P/256:GOS
UB6190: BY=P:GOSUB6190:GOSUB616
0
6100 IFZTHENBY=189+2*32:GOSUB61
80:GOSUB6160
6110 IFOP>=0THENBY=OP/256:GOSUB
6170:GOSUB6150:BY=OP:GOSUB6180
6120 IFR=0THEN5320
6130 GOSUB6160:BY=R:GOSUB6180:I
F(ABS(BORFNB(M,6))<>0)ANDO$<>"j
r"THENBY=R/256:GOSUB 6180
6140 GOTO5320
6150 IFZ<>0ANDBY<>0ANDABS((B=0)
*NOTB)<>0THENGOSUB6260:BY=VAL(F
$):GOSUB6180:Z=0
6160 PRINT" ";:RETURN
6170 IFINT(BY)<=0THENRETURN
6180 BY=FNJ(BY,8):POKEP,BY:P=P+
1
6190 BY=FNJ(BY,8):PRINTMID$(H$,
1+INT(BY/16),1);MID$(H$,FNJ(BY,
4)+1,1);
6200 RETURN
7000 IFK>NTHENI$="end":RETURN
7010 K1=K9+1:IFK9>=LEN(T$(K))TH
ENI$="/faltando/":RETURN
7020 K9=K1:IFMID$(T$(K),K1,1)="
 "THEN7010
7030 IFK9>LEN(T$(K))THENI$=RIGH
T$(T$(K),LEN(T$(K))-K1+1):RETUR
N
7040 IFMID$(T$(K),K9,1)<>" "THE
NK9=K9+1:GOTO7030
7050 I$=MID$(T$(K),K1,K9-K1):RE
TURN
8000 IFK>0THENIFRIGHT$(T$(K),LE
N(T$(K))-K9+1)>T$(99)THENPRINTR
IGHT$(T$(K),LEN(T$(K))-K9+1);
8010 K=K+1:K9=0
8020 PRINT:RETURN
9000 X$=""
9010 IFI$<"a"ORI$>"z"THEN9030
9020 X$=X$+LEFT$(I$,1):I$=RIGHT
$(I$,LEN(I$)-1):GOTO9010
9030 IFI$<>""THENRETURN
9400 FORG=1TOVV:IFFNE(X$,Z$(G))
THENRETURN
9410 NEXTG:VV=VV+1:Z$(VV)=X$:G=
VV:R(G)=23000
9420 RETURN
10420 CLS:LOCATE0,10:PRINT"Posi
cione a fita, pressione qualque
r   tecla e aperte PLAY no grav
ador"
10430 U$=INKEY$:IFU$=""THEN1043
0
10440 OPEN"CAS:ASM"FORINPUTAS#1
:CLS:LOCATE6,10:PRINT"Carregand
o programa fonte":INPUT#1,N:FOR
J=1TON:LINEINPUT#1,T$(J):NEXT:C
LOSE#1:RETURN
10450 CLS:LOCATE0,10:PRINT"Posi
cione a fita, aperte RECORD no
    gravador e pressione qualqu
er tecla"
10460 U$=INKEY$:IFU$=""THEN1046
0
10470 OPEN"CAS:ASM"FOROUTPUTAS#
1:CLS:LOCATE7,10:PRINT"Gravando
 programa fonte":PRINT#1,N:FORJ
=1TON:PRINT#1,T$(J):NEXT:CLOSE#
1:RETURN
10490 PRINT"Qual o número da li
nha (múltiplo de 10)";
10500 INPUTK:CLS
10510 K2=K/10:IFK2>NTHENK2=N+1:
N=N+1:T$(K2)=""
10520 IFK2<.1THENK2=.1
10530 IFK2=INT(K2)THEN10550
10540 K2=INT(K2)+1:FORK3=NTOK2-
1STEP-1:T$(K3+1)=T$(K3):NEXT:N=
N+1:T$(K2)=""
10550 P1=887:P0=P1
10560 LOCATE0,21:PRINTK;TAB(6)T
$(K2)+STRING$(10,32);:P9=P0+LEN
(T$(K2))
10570 IFP1<P0THENP1=P0
10580 IFP1>P9THENP1=P1-1
10590 VPOKEBASE(0)+P1-1,32:VPOK
EBASE(0)+P1,195
10600 P7=0:A$=INKEY$:IFA$=""THE
N10600
10610 IFA$=CHR$(13)THEN10700
10615 IFA$=CHR$(27)THENRETURN
10620 IFA$=CHR$(28)THENVPOKEBAS
E(0)+P1,32:P1=P1+1:GOTO10580
10630 IFA$=CHR$(29)THENVPOKEBAS
E(0)+P1,32:P1=P1-1:GOTO10570
10640 IFA$=CHR$(30)THENA$="":GO
TO10670
10650 IFA$=CHR$(31)THENA$=" "+M
ID$(T$(K2),P1-P0+1,1):P7=-1:GOT
O10670
10660 IFA$<" "THEN10600
10670 IFP1-P0+1>LEN(T$(K2))THEN
T$(K2)=LEFT$(T$(K2),P1-P0)+A$:G
OTO10690
10680 T$(K2)=LEFT$(T$(K2),P1-P0
)+A$+RIGHT$(T$(K2),LEN(T$(K2))-
P1+P0-1)
10690 P1=P1-(LEN(A$)>0)+P7:GOTO
10560
10700 LOCATE0,24:PRINT:K=K+10:G
OTO10510
10710 IFN=0THENCLS:PRINT" Nada
a apagar":RETURN
10720 CLS:PRINT"Qual o número d
a linha (múltiplo de 10)";
```

MICRO DICAS

COMO ENCONTRAR ERROS EM PROGRAMAS LONGOS

O erro mais comum em programas como o desta lição consiste em deixar de digitar um segmento de linha **DATA** (nisso também o MSX se assemelha ao ZX Spectrum). Assim, quando surge a mensagem "OUT OF DATA", devemos averiguar o que está faltando nas linhas **DATA**. Até mesmo a falta de uma vírgula causará problemas. Se o seu Assembler não funcionar na primeira vez, não se preocupe: o MSX possui uma função de rastreamento (*trace*) que o ajudará a encontrar os erros.

Para ativá-la, digite **TRON** no modo imediato, sem número de linha. Depois rode o programa usando **RUN**. O número da linha BASIC que estiver sendo executada nesse momento aparecerá na tela. Esse recurso torna mais fácil a detecção de erros nos programas mais longos.

A função é desativada pelo comando **TROFF**.

```
10730 INPUTK:K2=K/10
10740 IFK2>NORK2<1ORK2<>INT(K2)
THENPRINT"Esta linha não existe
":RETURN
10750 K=K2:FOR K3=K2 TO N:T$(K3
)=T$(K3+1):NEXT:N=N-1:LOCATE 0,
10:PRINT K*10;"  ";T$(K):RETURN
10760 IFN=0THENPRINT" Nada a li
star":RETURN
10770 PRINT"Quais os números da
 primeira e da      última linh
a (múltiplos de 10)";
10780 INPUTK,K2:K=INT(K):K2=INT
(K2):K1=K/10:K2=K2/10
10790 IFK2>NTHENK2=N
10800 IFK1<1THENK1=1
10810 IFK2<K1ANDK2=NTHENRETURN
10820 IFK2<K1THENCLS:PRINT"Núme
ros de linha inválidos":GOTO107
70
10830 CLS:FORK3=K1TOK2:PRINTK3*
10;"  ";T$(K3):NEXT
10840 RETURN
10900 END
```

Os espaços em branco foram eliminados de modo a diminuir a quantidade de memória ocupada pelo programa. Algumas linhas — 5450 e 5750, por exemplo — são quase do tamanho máximo permitido pelo computador; assim, qualquer espaço adicional pode fazer com que a porção final dessas linhas seja ignorada.

As linhas 5030, 5050, 5390, 5500 apresentam, cada uma, quatro caracteres em branco entre as aspas; as linhas 5660 e 5770, três; e as linhas 10750 e 10830, dois.

COMO FUNCIONA O PROGRAMA

Antes de montar o programa devemos proteger uma área de memória, onde o Assembler colocará os códigos por meio do comando **CLEAR**. Isso é feito na linha 5000, que reserva 500 bytes para os cordões (strings) e protege a memória a partir de E000 hexadecimal. Mude esse valor conforme as necessidades. Programas grandes podem exigir maior espaço de cordões.

Depois de digitar, rode o Assembler e verá um menu. Para digitar um programa em Assembly, tecle "e" (não se esqueça de destravar a tecla **CAPS**, para que o MSX aceite letras minúsculas). O computador pedirá então o número da linha. Cada instrução deve ser introduzida em uma linha numerada, seme-

lhante ao que ocorre no BASIC. É preciso que os números das linhas sejam múltiplos de 10; apenas um mnemônico Assembly com seus operandos pode ser introduzido em cada linha.

A primeira linha de seu programa deve ser mais ou menos assim:

```
10 org -8192
```

−8192 é o endereço inicial do programa em código. Obviamente, ele deve estar na área protegida da memória. Se esquecermos de definir **org**, ou seja, a origem, o Assembler colocará o programa a partir de DFFF, em hexadecimal (ou −8192, em decimal).

O programa aceita os códigos mnemônicos padrão do chip Zilog Z-80, com exceção dos comandos de retorno condicional. O comando de retorno incondicional de sub-rotinas, cujo mnemônico é **ret**, entretanto, *funciona* em nosso Assembler.

Algumas vezes, porém, podemos utilizar retornos condicionais, como **ret nz**, que significa "retorne se não for zero". Em nosso caso, use: **rts nz** (**rts** deve ser usado no lugar de **ret** sempre que precisarmos empregar retornos condicionais). Quando se trata de retorno incondicional — ou seja, **ret** não seguido de nenhuma letra — use o mnemônico padrão **ret**.

No artigo da página 213 (Programas em *Código de Máquina*), usamos o mnemônico **outir**, quando na verdade o padrão Z-80 é **otir**. Nosso Assembler aceita **otir**.

Números hexadecimais devem ser precedidos de **$**. Números binários, de **%**. Qualquer número sem nada na frente será interpretado como decimal. Qualquer expressão que não seja um comando Assembly será considerada como um rótulo *(label)*. Evite usar palavras parecidas ou muito longas; números não são permitidos.

Para que o Assembler saiba onde parar, termine seu programa com **end**.

Antes de apertar o **ENTER** para editar uma linha pressione as teclas do cursor. As setas "direita" e "esquerda" movem o cursor ao longo da linha; a seta "para baixo" serve para inserir caracteres e a seta "para cima", para apagá-los. A edição de linhas é bastante lenta e muitas vezes os caracteres digitados demoram a aparecer no vídeo.

Use a tecla **ESC** para sair do modo de edição. Essa tecla deve substituir o comando **ENTER**, depois que a última linha tiver sido digitada; caso contrário, uma linha em branco será incorporada ao programa em Assembly.

De volta ao menu, pressione "1" para obter uma listagem do programa em Assembly na tela.

Se houver algum erro na introdução do programa-fonte, volte ao menu e pressione "e"; forneça então o número da linha que deseja mudar.

Por outro lado, se quisermos inserir uma linha entre duas já existentes, bastará digitá-la usando um número intermediário, quando no modo de edição. Assim, ao listarmos o programa novamente, veremos que ele foi renumerado com a nova linha no lugar certo. Todavia, apenas uma linha pode ser introduzida dessa maneira de cada vez.

Para apagar uma linha, é preciso voltar ao menu, teclar "a", e fornecer o número da linha que queremos eliminar.

Quando estivermos satisfeitos com o programa Assembly a ser montado (*programa-fonte*), devemos voltar ao menu e teclar "m" para que ele seja "montado" na memória. Ao mesmo tempo, obteremos uma listagem dos códigos equivalentes na tela (*programa-objeto*).

Se a esta altura notarmos um erro em alguma das linhas, devemos voltar ao menu e editar a linha correspondente.

Uma vez montado o programa, o endereço final da rotina em código aparecerá na tela.

A opção de gravação — "g", no menu — armazena o programa-fonte em fita cassete — os mnemônicos, portanto. Para gravar o Assembler, use a via normal — **CSAVE**.

Para executar o programa em código, utilize instruções do tipo DEF **USRO** e **A=USRO**. Grave o programa-objeto, empregando:

```
BSAVE "CAS:nome", endereço
inicial, nº de bytes
```

"Nome" é a denominação do programa-objeto e deve estar entre aspas. **BSAVE** informa ao computador que está sendo gravado um programa em código e não em BASIC. O endereço inicial é a origem do seu programa na memória. Podemos calcular o número de bytes subtraindo a origem do endereço final e somando 1. Podemos ainda acrescentar o endereço inicial para execução (não mostrado no exemplo). Como esse endereço é quase sempre igual ao endereço inicial, sua inclusão é opcional.

UM TESTE

Para testar seu programa, entre o programa de deslocamento horizontal da tela para a direita que se encontra na página 215. Uma vez traduzido para código, ele ficará assim:

```
11 C0 E4 21 BF E4 6 18 C5 1A 01
 27 00 ED B8 12 2B 1B C1 10 F3
C9
```

Note que devemos usar letras minúsculas para digitar o programa-fonte. Nosso Assembler não reconhece comandos em maiúsculas.

GRÁFICOS INSTANTÂNEOS

Você não precisa conhecer código de máquina nem entender de sistema binário para animar seus jogos com figuras simples. Leia este artigo e veja como é fácil fazer isso.

Qualquer pessoa com algumas horas de experiência com computadores pode criar caracteres gráficos originais para usar em jogos. Tudo o que se precisa é lápis e papel para esquematizar as idéias, além de duas (ou, no máximo, três) rotinas simples para transformar tais idéias em figuras no computador.

Cada tipo de máquina exige um método particular para a criação de novos padrões de caracteres gráficos definidos pelo usuário. O tamanho desses caracteres também é variável. O MSX tem sprites de 16 por 16 pixels (ou pontos), enquanto o Spectrum oferece apenas um UDG de 8 por 8 pixels (os compatíveis com o ZX-81 não serão tratados aqui).

No entanto, independentemente do que seu computador ofereça, o mais conveniente é começar criando figuras de 8 x 8 (tamanho aproximado de um "inimigo" em um jogo espacial). Uma vez que você tenha aprendido a trabalhar com esse padrão, será fácil criar figuras maiores, ou mesmo encadear duas ou mais pequenas figuras para formar uma terceira.

DO DESENHO PARA O BINÁRIO

Em alguns computadores, os dados necessários para definir um UDG podem ser fornecidos diretamente na forma binária. Em outros, você terá que convertê-los em decimal ou em hexadecimal. Assim, leia primeiramente a seção para sua máquina antes de fazer qualquer conversão. No Apple você terá que usar o nosso editor de figuras.

Os computadores armazenam as informações que lhes são dadas em binário. Mas você não precisa conhecer esse sistema numérico para transformar seus desenhos em filas de números binários. Tudo o que deve fazer para isso é:

— Utilizar o número 1 sempre que quiser obter um ponto.
— Empregar o número 0 toda vez que quiser um espaço.

Tomemos como exemplo o padrão da cruz de Lorena. A linha do topo é composta de três espaços, um ponto e mais quatro espaços. Em binário, 00010000.

A segunda linha é formada por dois espaços, três pontos e três espaços: 00111000. Todo o padrão pode ser representado assim:

```
0 0 0 1 0 0 0 0
0 0 1 1 1 0 0 0
0 0 0 1 0 0 0 0
0 1 1 1 1 1 0 0
0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
```

DE BINÁRIO PARA DECIMAL

A maneira mais rápida de converter binário em decimal é usar a pequena tabela de nove linhas por oito colunas que apresentamos a seguir. Na primeira linha, escreva os números 128, 64, 32, 16, 8, 4, 2, 1. Nas outras oito coloque os números correspondentes ao gráfico que você quer. Aqui temos, por exemplo, a tabela para a cruz de Lorena:

```
128  64  32  16   8   4   2   1
  0   0   0   1   0   0   0   0
  0   0   1   1   1   0   0   0
  0   0   0   1   0   0   0   0
  0   1   1   1   1   1   0   0
  0   0   0   1   0   0   0   0
  0   0   0   1   0   0   0   0
  0   0   0   1   0   0   0   0
  0   0   0   0   0   0   0   0
```

Para fazer a conversão, ignore os zeros. Inicialmente, compare os 1 em binário com o número no topo da coluna. Some então todos os números correspondentes aos 1 da primeira linha. Repita essa operação em todas as linhas.

No exemplo, a primeira linha contém o número 1 apenas na quarta coluna. Portanto, temos 16.

A segunda linha tem três 1 que correspondem a 32, 16 e 8. Soma: 56.

Quando o processo de verificação houver terminado, você terá oito números decimais. A declaração **DATA** necessária para entrar os dados no computador ficará assim:

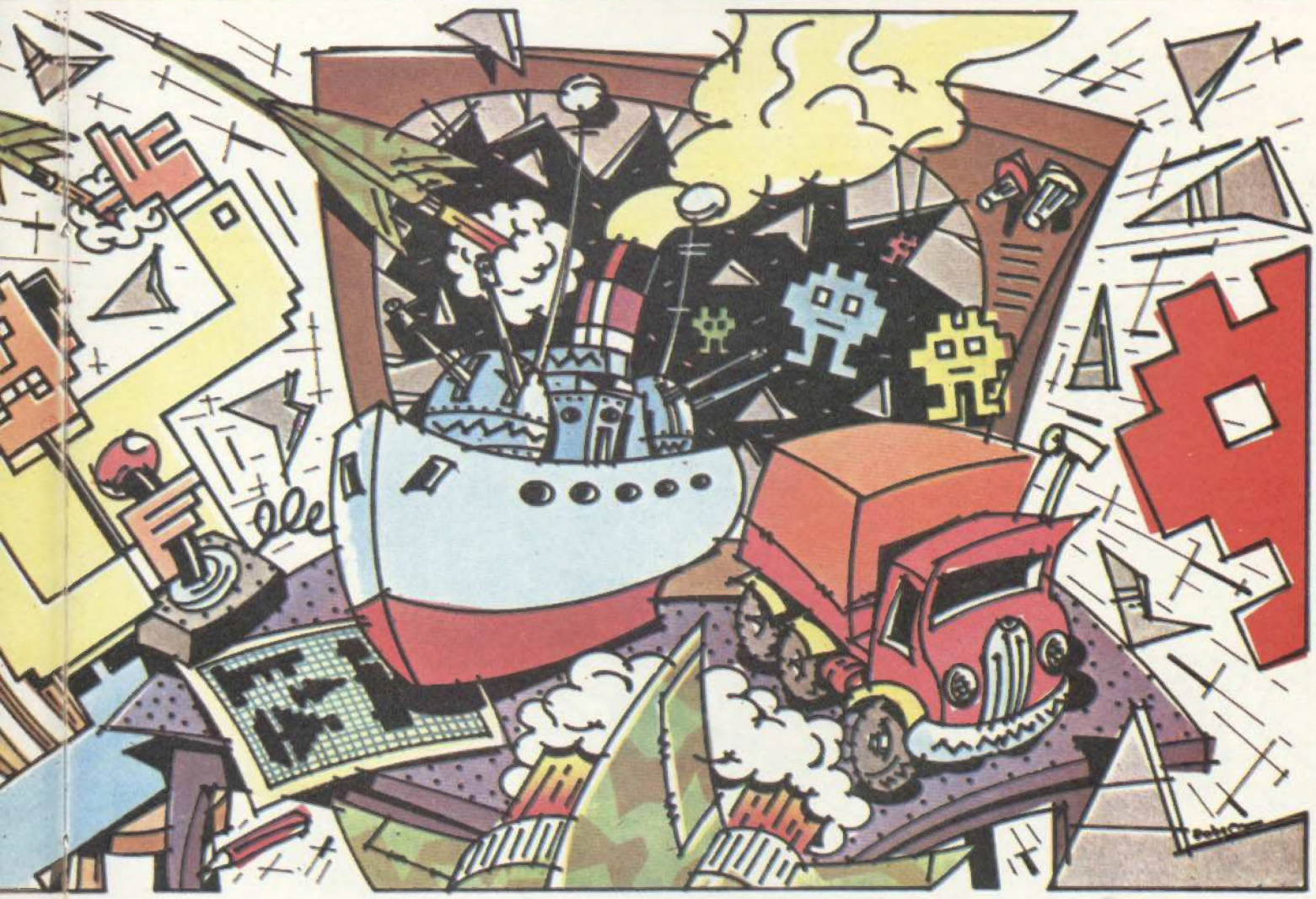
```
DATA 16,56,16,124,16,16,16,0
```

De início, você pode ficar com a impressão de que esta é uma forma muito complicada de fazer o trabalho. No entanto, depois de meia dúzia de tentativas, seu ritmo de trabalho se tornará suficientemente rápido. Além disso, as combinações mais comuns, como 255 (ou 11111111 em binário), serão logo memorizadas, poupando-as da necessidade de fazer cálculos para chegar ao sistema decimal.

DE BINÁRIO PARA HEXADECIMAL

Converter números binários em he-

- TRANSFORME DESENHOS EM FILAS DE NÚMEROS BINÁRIOS
- CONVERSÕES DE BINÁRIO PARA DECIMAL E PARA HEXADECIMAL
- USE GRÁFICOS EM PROGRAMAS



xadecimais é ainda mais fácil. Para isso, basta aplicar a tabela a seguir, não sendo necessário nem mesmo escrever nela os números binários:

Binário	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Desta vez não ignore os zeros. Para começar, tome os primeiros quatro dígitos do número binário e procure o seu equivalente em hexa. Em seguida, faça o mesmo com os quatro dígitos seguintes. A reunião dos dois dígitos hexadecimais obtidos define o número que você precisa.

Voltando à cruz de Lorena, a primeira metade do binário da primeira linha é 0001. Em hexa temos, pela tabela, 1. A segunda metade é 0000. Em hexa, temos 0. Escreva os dois números juntos e teremos 10, o hexa correspondente ao binário 00010000.

De forma similar, na segunda linha temos 0011 (que é 3 em hexa) e 1000 (8 em hexadecimal). Assim, o número hexadecimal completo é 38.

Repita o processo até a última linha e terá oito números hexadecimais. A sua declaração **DATA** completa ficará como segue:

```
DATA 10,38,10,7C,10,10,10,00
```

Além disso, a única coisa a fazer é dizer ao computador (que não pode adivinhar) se o número que você está digitando é decimal ou hexa. Veja como fazer isto na seção específica do seu computador.

Evidentemente, é possível escrever um pequeno programa para fazer a conversão de bases numéricas, mas ele não será muito útil se você quiser modificar alguma informação que já esteja na memória do computador.

T

O TRS-Color aceita **DATA** em decimal, hexa ou binário.

Se os dados (**DATA**) estão em hexa, você terá que adicionar uma linha extra ao programa para convertê-los em decimal (veja a seguir).

O primeiro programa desenha um pequeno avião no canto superior esquerdo do vídeo. Este pode não ser o melhor lugar para vê-lo; mas é o mais fácil para começar, se você quiser fazer um programa para movimentá-lo pela tela.

```
20 PMODE 4,1
30 PCLS
40 SCREEN 1,1
60 FOR L=0 TO 7
70 READ N$
80 POKE L*32+1536,VAL("&H"+N$)
90 NEXT L
110 GOTO 110
500 DATA 00,10,18,9C,FF,9C,18,1
0
```

Digite o programa e execute-o.

PMODE 4,1 foi escolhido na linha 20 porque só se pode produzir gráficos UDG quando neste modo de resolução gráfica mais alta.

Para limpar a tela sobre a qual vai se desenhar, deve-se usar o comando **PCLS** como na linha 30. Ele se aplica não só ao **PMODE 4,1**, como a todos os modos de alta resolução.

Use **SCREEN 1,1**, como na linha 40, para ligar a tela de alta resolução de forma a mostrar o seu UDG. **SCREEN 1,1** também seleciona as cores branca e preta.

O laço **FOR...NEXT** das linhas 60 e 90 faz com que a linha 70 seja executada oito vezes. Toda vez que encontra **READ N$**, o computador lê o dado seguinte da linha **DATA** de número 500.

A linha 80 é importante por duas razões. A primeira é que ela faz com que o padrão de pontos definido pela linha 500 apareça no vídeo. A segunda é que ela converte os números hexadecimais da linha **DATA** em decimais e coloca-os diretamente na área de memória que controla o que aparece na tela do computador.

Finalmente, a linha 110 é um laço que mantém a alta resolução. Sem essa linha o programa terminaria, provocando o retorno automático para o modo texto; em conseqüência, você não veria o seu desenho na tela.

GRÁFICOS UDG MAIS ALTOS

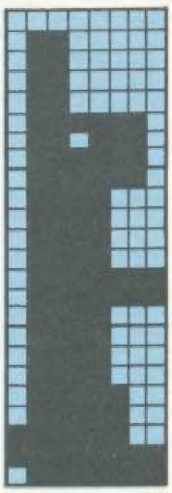
Você pode criar um UDG "alto e magro", em vez de um 8x8, mudando a linha 60 e alterando os dados da linha 500. Em primeiro lugar, modifique a linha 60 para:

```
60 FOR L=0 TO 7
```

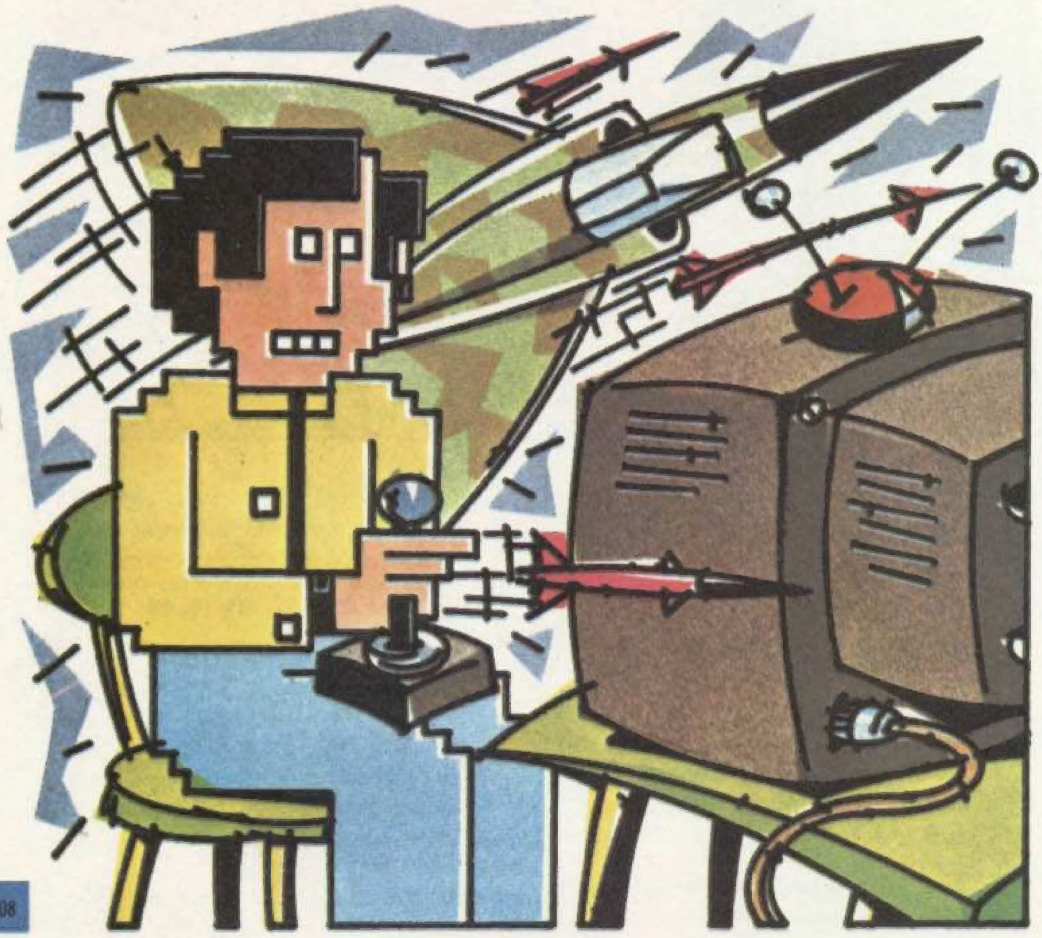
Em seguida, mude a linha 500 de forma que fique assim:

```
500 DATA 00,10,18,9C,FF,9C,18,1
0
```

A alteração na linha 60 permite a leitura de um maior número de dados da linha 500. E esta define a imagem de um coelho, usando hexadecimais. Assim, ao executar o programa, você encontrará a figura do coelho na tela.



Esse método permite a criação de gráficos de qualquer tamanho. Se você já desenhou a sua figura em papel quadriculado, conte quantas linhas foram usadas. Altere a linha 60 usando o número de linhas do desenho. A seguir, faça com que o número de informações da linha **DATA** corresponda ao número de vezes que o laço **FOR...NEXT** for executado.



GRÁFICOS UDG MAIS LARGOS

O desenho de figuras baixas e largas (em vez de altas e estreitas) exige um pouco mais de trabalho.

Comece mudando as linhas 60 e 80 da seguinte forma:

```
60 FOR L=0 TO 7
80 POKE L*32+1536+F,VAL("&H"+N$
```

Altere a linha **DATA** de forma que apareça assim:

```
500 DATA 0F,0F,EF,EF,EF,EF,FE,4
4,FF,FF,FF,FF,FF,FF,40,00,FF,FF
,FF,FF,FF,FF,66,66
```

Finalmente, adicione estas linhas:

```
50 FOR F=0 TO 2
100 NEXT F
```

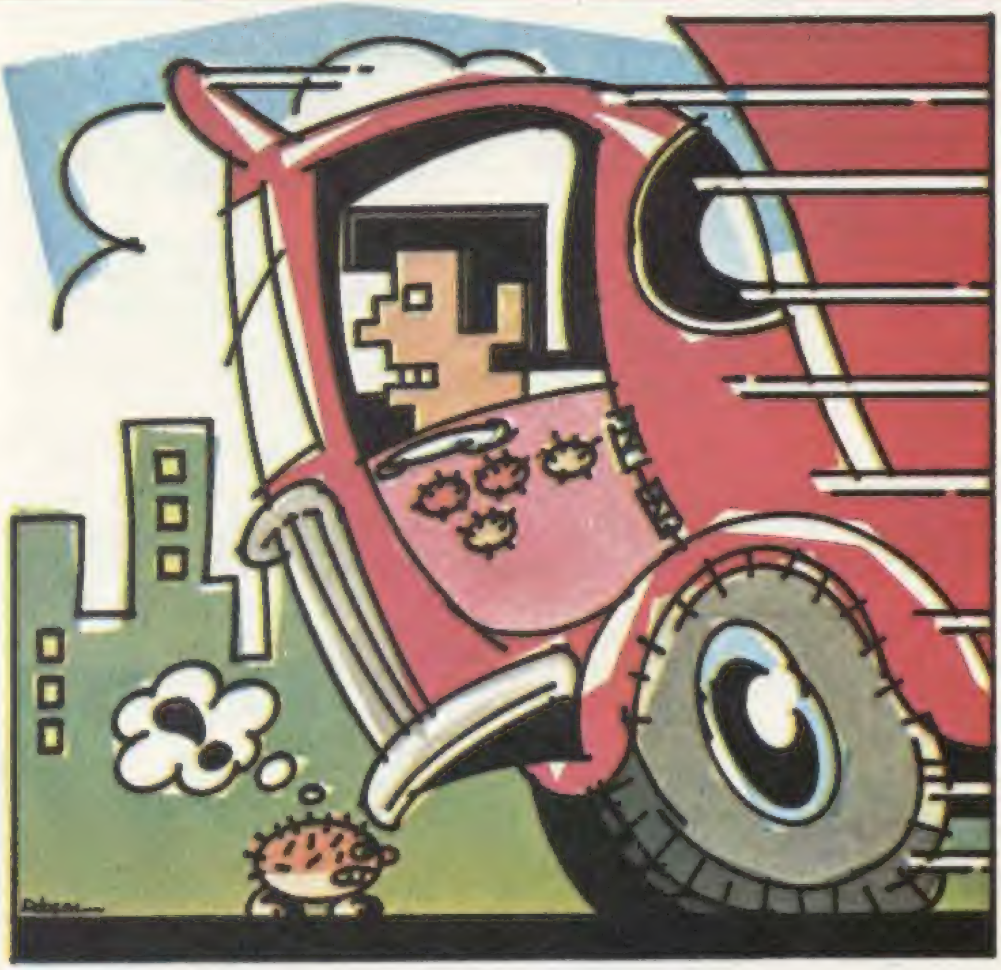
Quando você executar o programa, verá na tela a figura de um caminhão de carga.

Como isto funciona? Os 24 trechos de informação da linha 500 formam três quadrados de oito pixels de lado. Se os dados fossem lidos por um único **FOR...NEXT** como antes, o caminhão apareceria de forma peculiar, cortado em três pedaços que se amontariam no vídeo. Assim, o computador deve ser avisado para colocar os blocos lado a lado. Para fazer isso, um laço extra altera o valor do **POKE** na linha 80, de forma que o segundo bloco apareça na linha do topo, ao lado do primeiro. Depois de ser lido, o terceiro bloco é colocado ao lado dos outros dois por intermédio do comando **POKE**.

MOVIMENTE OS DESENHOS

Você pode mover a figura do avião pela tela usando estas linhas:

```
110 DIM A(3),B(3)
120 GET (0,0)-(7,7),A,G
130 PCLS
140 LET X=127
150 LET Y=95
160 PUT (X,Y)-(X+7,Y+7),A,PSET
170 LET LX=X
180 LET LY=Y
190 IF PEEK(338)=251 AND Y>2 TH
EN Y=Y-2:GOTO 240
200 IF PEEK(342)=253 AND Y<182
THEN Y=Y+2:GOTO 240
210 IF PEEK(340)=247 AND X>3 TH
EN X=X-3:GOTO 240
220 IF PEEK(338)=247 AND X<245
THEN X=X+3:GOTO 240
230 GOTO 190
240 PUT (LX,LY)-(LX+7,LY+7),B,P
SET
250 GOTO 160
```



Quando você executar este programa, a tecla Z movimentará a figura para a esquerda; X o fará para a direita; P para cima; e L para baixo.

O comando **GET** permite ao computador lembrar-se do que está em um determinado ponto da tela, enquanto **PUT** coloca o que foi encontrado em qualquer outro lugar. **DIM**, por sua vez, reserva espaço na memória para **GET**.

As linhas 190 até 220 verificam se uma tecla foi pressionada e movem o UDG em caso positivo.

Se você quiser mover o coelho deve fazer as seguintes alterações:

```
110 DIM A(6),B(6)
120 GET (0,0)-(7,23),A,G
160 PUT (X,Y)-(X+7,Y+23),A,PSET
200 IF PEEK(342)=253 AND Y<166
THEN Y=Y+2:GOTO 240
240 PUT (LX,LY)-(LX+7,LY+23),B,
PSET
```

Para movimentar o caminhão, faça estas modificações:

```
110 DIM A(6),B(6)
120 GET (0,0)-(23,7),A,G
160 PUT (X,Y)-(X+23,Y+7),A,PSET
200 IF PEEK(342)=253 AND Y<182
THEN Y=Y+2:GOTO 240
220 IF PEEK(338)=247 AND X<229
THEN X=X+3:GOTO 240
240 PUT (LX,LY)-(LX+23,LY+7),B,
PSET
```

COMO USAR DADOS EM BINÁRIO

Você pode colocar números em binário nas linhas **DATA**, se quiser, mas certifique-se antes de que cada número consiste de um byte de oito bits. Você terá, também, que adicionar estas linhas ao programa:

```
71 LET N=0
72 FOR J=1 TO 8
74 IF MID$(N$,J,1)="1" THEN N=N
+2^(8-J)
76 NEXT J

80 POKE L*32+1536+F,N
```

As novas linhas examinam os bytes bit por bit, transformando-os em números decimais. Estes são, resumidamente, equivalentes aos da tabela de conversão mostrada na introdução deste artigo.

A movimentação de figuras em alta

resolução será apresentada em detalhes num artigo próximo.

No Apple, entretanto, a geração de figuras em alta resolução não segue os mesmos padrões de outros computadores, sendo muito mais complicada (em artigo anterior, publicamos um editor de figuras móveis que permite aos usuários desse microcomputador utilizarem figuras compostas por blocos de 8x8 pontos).

Neste artigo você poderá treinar a utilização do editor de figuras. Os programas a seguir só funcionarão se as figuras correspondentes tiverem sido geradas pelo editor.

Carregue o programa editor de figuras no seu computador e coloque o desenho do pequeno monstro a seguir em linhas **DATA** usando asteriscos.

Execute o programa editor e a figura será armazenada na memória. Feito isso, digite **NEW** e o seguinte programa:

```
10  HOME : HGR
20  SCALE= 1: ROT= 0
30 X = 140:Y = 90
40  GOTO 200
50 LX = X:LY = Y: GET A$
60 A =  ASC (A$)
70  IF A = 80 AND Y > 8 THEN Y
= Y - 4: GOTO 200
80  IF A = 76 AND Y < 150 THEN
Y = Y + 4: GOTO 200
90  IF A = 90 AND X > 8 THEN X
= X - 4: GOTO 200
100  IF A = 88 AND X < 270 THEN
 X = X + 4: GOTO 200
110  GOTO 50
200  HCOLOR= 0
210  DRAW 1 AT LX,LY
220  HCOLOR= 3
230  DRAW 1 AT X,Y
240  GOTO 50
```

Não há necessidade de copiar as linhas **DATA** geradas pelo programa editor. Uma vez criada, a figura permanecerá protegida no topo da memória até que o computador seja desligado, ou até que o programa editor crie outra forma. Nem mesmo **NEW** pode destruí-la.

O programa movimenta a figura com o auxílio das teclas P, L, X e Z. As linhas 50 a 110 fazem isso da maneira usual.

As linhas de 10 a 40 cuidam das condições iniciais: limpeza e definição da tela, escala, orientação e posição inicial da figura.

As linhas de 200 a 240 apagam o monstro de sua última posição, dada por LX e LY, e o desenham na nova, dada por X e Y.

DESENHOS MAIS LARGOS

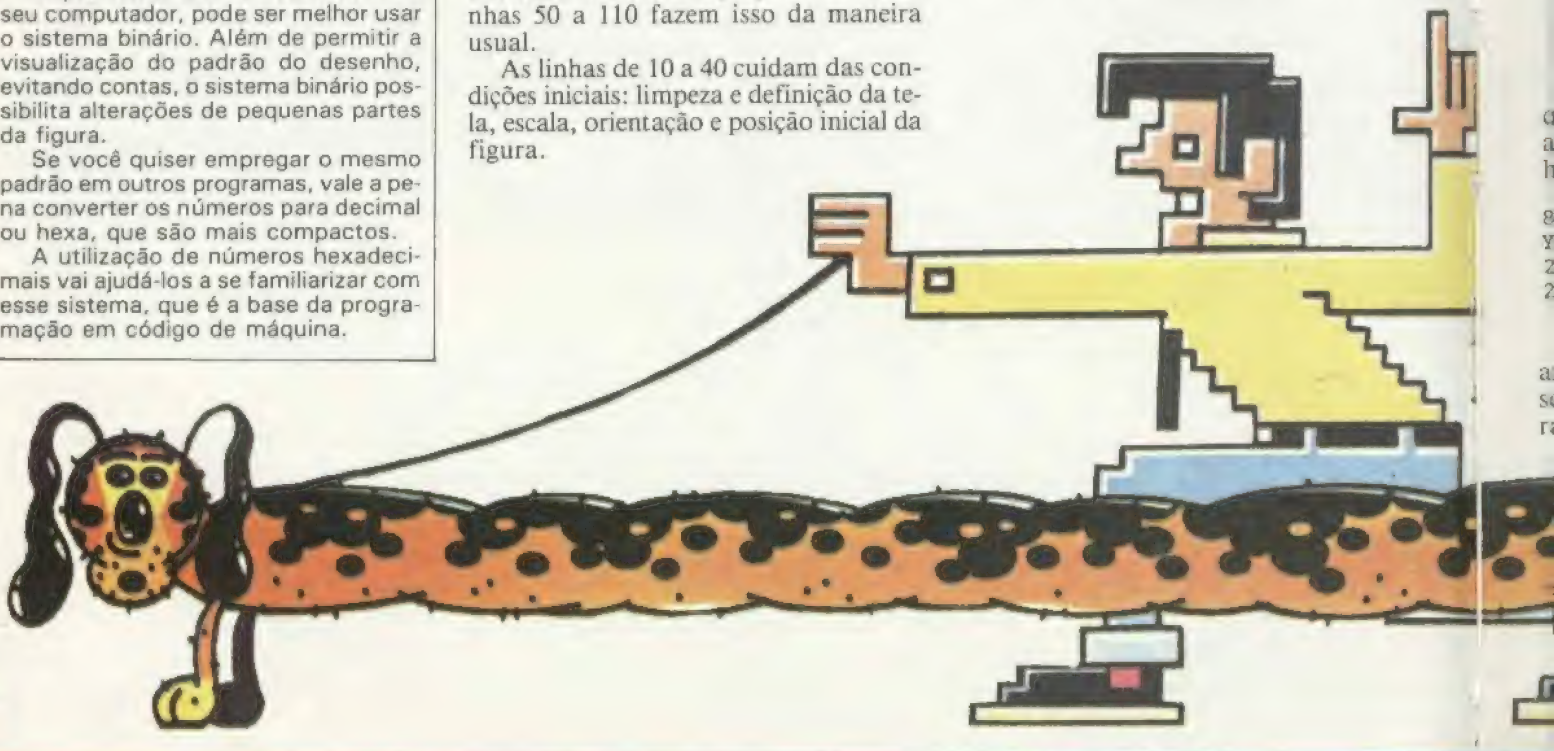
Desenhos compostos de mais de um bloco de 8x8 pontos exigem um cuidado especial com as coordenadas. Esse cuidado é necessário para que as figuras se posicionem de acordo com o planejado. Para familiarizar-se com esse tipo de desenho, gere os dois blocos que compõem o cachorro com o auxílio do programa editor.

Feito isso, digite **NEW** e carregue o programa anterior. Faça as seguintes modificações para mover o cachorro:

```
100  IF A = 88 AND X < 260 THEN
 X = X + 4: GOTO 200
215  DRAW 2 AT LX + 8,LY
235  DRAW 2 AT X + 8,Y
```

A condição após **AND** na linha 100 evita que o desenho ultrapasse a margem direita da tela. Ela teve que ser modificada porque agora o desenho é mais largo.

Como a figura tem dois blocos de 8x8

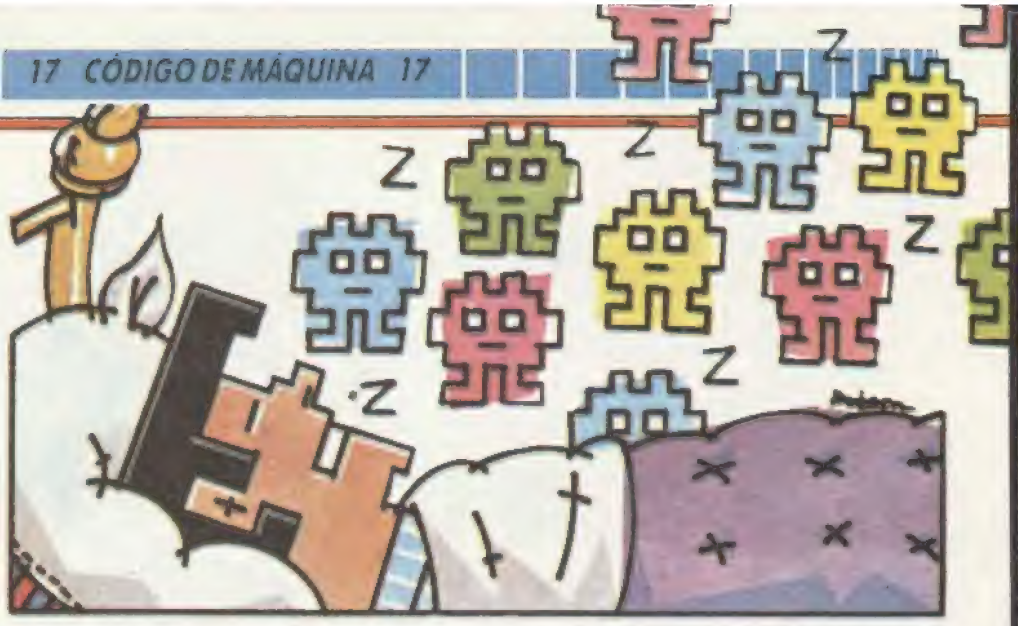


Que sistema de numeração — binário, decimal ou hexa — é melhor usar nas linhas *DATA*?

Dependendo das características do seu computador, pode ser melhor usar o sistema binário. Além de permitir a visualização do padrão do desenho, evitando contas, o sistema binário possibilita alterações de pequenas partes da figura.

Se você quiser empregar o mesmo padrão em outros programas, vale a pena converter os números para decimal ou hexa, que são mais compactos.

A utilização de números hexadecimais vai ajudá-los a se familiarizar com esse sistema, que é a base da programação em código de máquina.

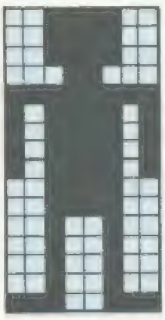


pontos, foi necessário colocar duas novas linhas com comandos **DRAW**, uma para desenhar e outra para apagar o segundo bloco.

É importante notar como são desenhados lado a lado os blocos que compõem o cachorro, ou seja, da esquerda para a direita, e de cima para baixo. Assim, as coordenadas no comando **DRAW** se referem ao canto superior esquerdo de cada bloco. Se o primeiro bloco está desenhado nas coordenadas X, Y, e queremos que o segundo seja colocado à sua direita, a posição deste último deve ser X + 8,Y. Isso acontece porque o canto superior esquerdo do segundo bloco deve ficar na mesma linha do canto superior esquerdo do primeiro (mesma coordenada vertical Y) oito pontos mais à direita.

UMA FIGURA ALTA E MAGRA

Continuando nosso treino, carregue novamente o editor e gere os dois blocos que desenham a silhueta de um homem.



Digite **NEW** e carregue o programa que movimenta o monstro. Faça então as seguintes modificações para mover o homem.

```
80  IF A = 76 AND Y < 140 THEN
Y = Y + 4: GOTO 200
215  DRAW 2 AT LX,LY + 8
235  DRAW 2 AT X,Y + 8
```

Da mesma maneira que no exemplo anterior, a condição na linha 80 teve que ser modificada para que o desenho, agora mais alto, não ultrapasse a margem inferior da tela.

As linhas 215 e 235 apagam e desenham, respectivamente, o segundo bloco. Se quisermos colocá-lo abaixo do primeiro bloco, teremos que situá-lo na mesma coluna, oito pontos mais embaixo. Assim, se a posição do primeiro bloco for X, Y, a do segundo deverá ser X,Y + 8.

Embora o MSX possua caracteres definíveis pelo usuário (com 8x8 pontos, como nas outras máquinas), geralmente é preferível utilizar os sprites, que são mais versáteis e mais adequados à movimentação de figuras.

Todavia, existem situações em que esses caracteres podem ser muito úteis. Uma aplicação muito comum é a redefinição total ou parcial do conjunto de caracteres. Isso às vezes é desejável em programas que devem utilizar alfabetos estrangeiros ou caracteres gráficos diferentes.

O conjunto de caracteres do MSX é muito completo, contendo todo o alfabeto grego e uma infinidade de outros caracteres gráficos. Falta-lhe, porém, a letra *è* (com acento grave), utilizada na língua francesa.

Para criar esse caractere, aplique o mesmo método já descrito neste artigo. A tabela que se segue traz a codificação do padrão do caractere desejado, pronta para ser colocada em uma linha **DATA**.

Binário	Hex	Decimal
00010000	10	16
00001000	08	8
00111100	3C	60
01100110	66	102
01111110	7E	126
01100000	60	96
00111100	3C	60
00000000	00	0

O formato decimal é mais fácil de usar. Mas mais difícil de calcular. O MSX aceita números binários e hexadecimais, desde que certos padrões sejam respeitados.

Para incorporar o novo caractere, use o programa:

```
10 SCREEN 0:KEY OFF
20 FOR I=0 TO 959
30 VPOKE BASE(0)+I,205
40 NEXT
```

```
50 E=205*8
60 FORI=E TO E+7
70 READ B$
80 VPOKE BASE(2)+I,VAL("&B"+B$)
90 NEXT I
100 FOR I=1 TO 2000:NEXT I
110 CLS:END
500 DATA 00010000
501 DATA 00001000
502 DATA 00111100
503 DATA 01100110
504 DATA 01111110
505 DATA 01100000
506 DATA 00111100
507 DATA 00000000
```

Ao ser executado o programa, a tela é ocupada por um símbolo gráfico. Em seguida esse símbolo se transforma no caractere que procuramos, ou seja, a letre è com acento grave. Após alguns instantes, a tela fica novamente limpa. O novo caractere, contudo, continua na memória e pode ser obtido da mesma maneira que o símbolo gráfico do qual tomou o lugar. Para isso, basta pressionar as teclas **GRAPH** e "E" ao mesmo tempo. O caractere só desaparecerá da memória quando utilizarmos o comando **SCREEN** ou quando desligarmos o computador.

ENTENDA O PROGRAMA

A linha 10 escolhe a tela de textos com 40 colunas e apaga do rodapé da tela as teclas de função. O laço das linhas 20 a 40 enche a tela com o caractere de código igual a 205 — correspondente a **GRAPH + E**.

Os padrões de todo o conjunto de caracteres estão guardados na VRAM — memória de vídeo — a partir do endereço dado por **BASE (2)**. Para saber como é o formato de um caractere, o computador multiplica o valor de seu código por 8 e soma esse valor ao de **BASE (2)**. O número que resultar dessa conta será o endereço do primeiro de uma série de oito bytes onde ficará armazenada a forma do caractere.

A linha 50 multiplica por 8 o código do caractere que vai ser modificado, para saber onde seu formato está na VRAM. O laço das linhas 70 a 90 coloca nesse local o padrão do novo caractere; os números são lidos com **READ** nas linhas **DATA**. Na linha 80, **VAL (&B + B$)** permite que sejam utilizados números binários.

A linha 100 retarda o processo até que a linha 110 apague a tela e termine o programa.

Se for preciso economizar espaço, as linhas **DATA** podem ser substituídas por uma única linha.

```
500 DATA 00010000,00001000,0011
1100,01100110,01111110,01100000
,00111100,00000000
```

Essa linha única, porém, não permite que se visualize claramente o caractere.

Se quisermos usar números hexadecimais, devemos fazer as seguintes modificações:

```
80 VPOKE BASE(2)+I,VAL("&H"+B$)
500 DATA 10,08,3C,66,7E,60,3C,0
0
```

Uma linha **DATA** desse tipo é bem mais fácil de ser digitada.

Para criar o mesmo caractere na tela de textos com 32 colunas, faça as seguintes modificações:

```
20 FOR I=0 TO 767
30 VPOKE BASE(5)+I,205
80 VPOKE BASE(7)+I,VAL("&H"+B$)
```

Note que a linha **DATA** deve estar em hexa.

MICRO DICAS

PROGRAMAS EDITORES

Crie suas próprias figuras no Apple e no MSX, utilizando os programas editores fornecidos por *INPUT* para estes computadores. Eles calcularão os números nas linhas **DATA** para você.

Lembre-se de que não será preciso digitar essas linhas. Ambos os computadores possuem facilidades de edição que permitem usar a própria tela produzida pelo editor — uma vez interrompido o programa — para criar uma linha **DATA** com um número alto na frente (a partir de 5000).

No MSX isto é feito com as teclas do cursor, **INS**, **DEL** e **ENTER**. No Apple, com **ESC**, **I**, **J**, **K**, **M**, setas e **RETURN**.

No Apple e no TK-2000 pode-se gravar a tabela de figuras usando o comando **W** do monitor. Quem tiver disco, pode usar o comando **BSAVE**.

SPECTRUM

O Spectrum aceita dados tanto em binário quanto em decimal, mas não em hexa. Para ver como ele trabalha com gráficos, digite:

```
PRINT  (graphics A)"
```

Para poder obter o símbolo (graphics A), você deverá pressionar <**CAPS SHIFT**> e <**GRAPHICS**> ao mesmo tempo e depois teclar "A".

O que você vê parece um A maiúsculo normal. Mas você pode defini-lo como uma forma de 8x8 pontos.

Para fazer isto, tudo o que você precisa é de um programa de cinco linhas. Assim, se você quiser plantar o pinheiro da ilustração, deve digitar as linhas a seguir:

```
10 FOR n=0 TO 7
20 READ data
30 DATA BIN 00010000,BIN 0001
1000,BIN 00111000,BIN 0011110
0,BIN 01111100,BIN 01111110,
BIN 11111110,BIN 00010000
40 POKE USR "a"+n,data
50 NEXT n
```

Esse programa usa um laço **FOR...NEXT** nas linhas 10 e 50 para dar entrada às oito linhas do desenho em seqüência. A linha 20 diz ao computador para ler dados na linha **DATA** e a linha 40 coloca esses dados na memória do computador, usando o comando **POKE**.

Execute o programa e digite:

```
PRINT "(graphics A)"
```

Você verá que o "A" desapareceu e em seu lugar aparece um pinheiro. Se você digitar **NEW** agora, o programa será apagado da memória, mas o desenho ficará na memória até que o computador seja desligado. Assim, você pode movê-lo pela tela ou usá-lo para efeitos decorativos como se ele fosse um caractere qualquer. Tente este exemplo:

```
 5 CLS
10 FOR y=3 TO 19
20 LET x=INT (RND*20)+5
30 PRINT AT y,x; INK 4;"(grap
hics A)"
40 LET xx=INT (RND*20)+5
50 PRINT AT y,xx; INK 2;"(gra
phics A)"
60 NEXT y
```

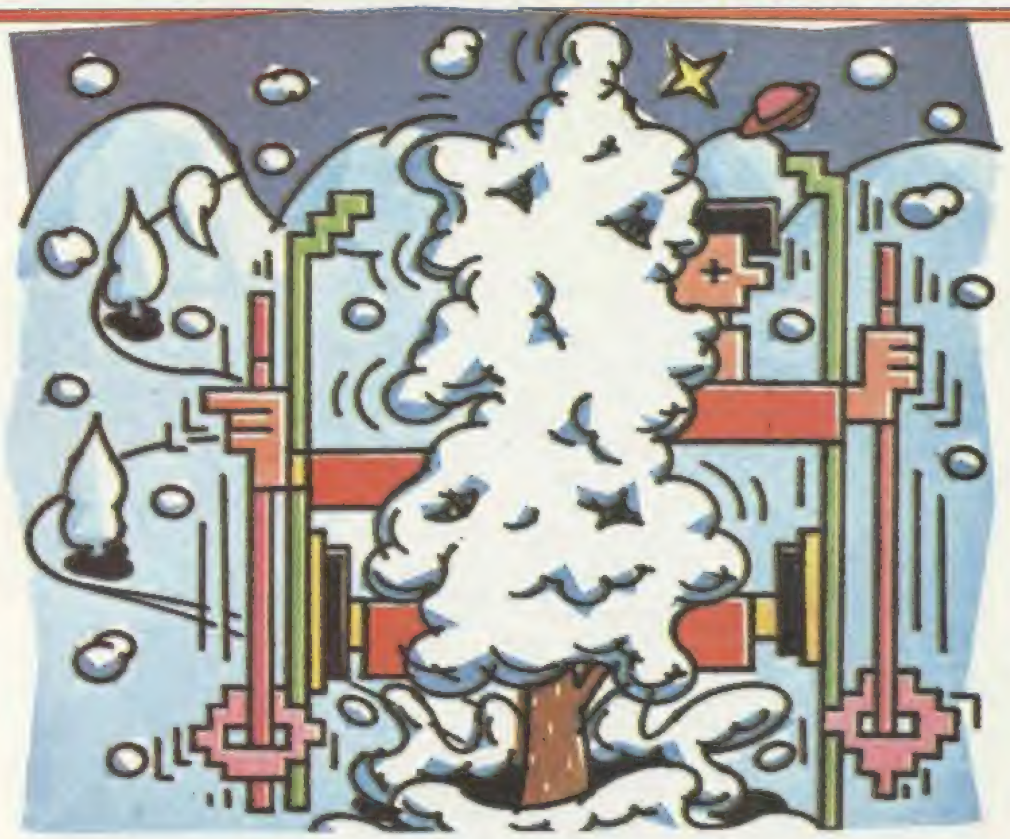
Esses pinheiros poderiam ser usados como obstáculos para um jogo de esqui.

DESENHE UM NAVIO

Quando você quiser criar gráficos maiores que 8x8, desenhe dois ou mais blocos de 8x8.

O programa a seguir, por exemplo, cria a proa de um destróier (que você pode colocar em um de seus jogos, usando as técnicas apresentadas em *Programação de Jogos*):

```
10 FOR n=0 TO 7
20 READ a
30 DATA BIN 0,BIN 0,BIN 0,BIN
00000111,BIN 00000011,BIN 111
11111,BIN 00000011,BIN 111111
11,BIN 01111111,BIN 00111111
40 POKE USR "a"+n,a
50 NEXT n
```



Dois aspectos devem ser ressaltados. Primeiro: você pode usar **BIN 0** se a linha inteira for vazia. Segundo: coloque na linha 20 uma variável simples "a" em lugar de "data", que foi inserida no programa anterior para que este ficasse mais compreensível. Poderia ser qualquer outra coisa, como "b", "c" ou "x", ou mesmo "maria"; mas para isso deveria ser utilizada uma variável idêntica na linha 40.

Quando você for entrar os dados para as outras duas partes do navio, não haverá necessidade de redigitar o programa todo. Depois de executá-lo a primeira vez, e com o primeiro gráfico na memória, mude a linha 40 para **USR "b"**, por exemplo. Mude também a linha 30 de modo que ela passe a conter os novos dados das outras partes do navio.

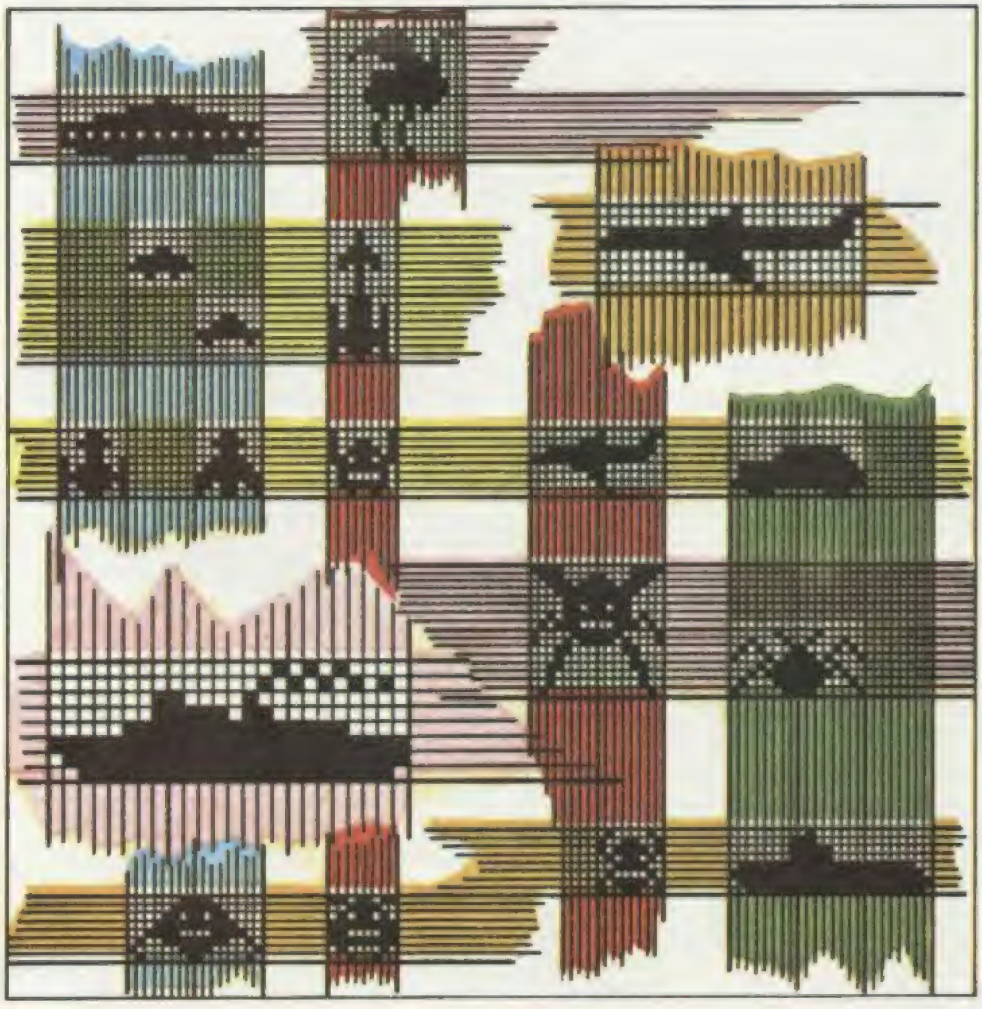
Tome cuidado para que as suas linhas **DATA** tenham sempre oito dados, mesmo que estes sejam apenas zeros. Um número menor de linhas provocará um erro do tipo: E Out of DATA, 20:1. Coloque linhas a mais e você descobrirá que seu navio está afundando.

Finalmente, para entrar os dados em decimal, converta os binários de acordo com o método já descrito; a seguir, coloque-os na linha **DATA** omitindo a palavra **BIN**. A linha 30 do programa do destróier ficará assim:

```
30 DATA 0,0,0,7,3,255,127,63
```

E AGORA...

Uma vez compreendidos os princípios gerais, você pode colocar no computador qualquer um dos gráficos desta página. Depois de ter feito uns três deles, você verá que já tem prática suficiente para criar os seus próprios desenhos e colocá-los no computador.



UM ASSISTENTE DE ARTE

Mesmo não sendo um Leonardo da Vinci, você pode realizar belos desenhos com a ajuda do computador. Para isso, basta executar o programa apresentado nesta lição.

Você não precisa ser um Rafael ou mesmo um Pablo Picasso para desenhar com desenvoltura. Devidamente programado, seu computador pode se transformar em uma poderosa ferramenta de desenho, capacitando-o a realizar trabalhos gráficos de boa qualidade com liberdade igual (ou até maior) que a de um artista que trabalha apenas com material convencional: lápis e papel.

Na indústria, a combinação de programas de Desenho Auxiliado por Computador (conhecido pela sigla CAD, da expressão em língua inglesa *Computer Aided Design*) com computadores de grande porte permite não só a execução de desenhos de objetos com muita riqueza de detalhes como também a simulação de situações em que tais objetos são submetidos à ação de determinados fatores como carga, vento, vibração ou variações térmicas. Essas técnicas extrapolam a capacidade do micro doméstico, visto que exigem o processamento de um volume muito grande de dados, assim como uma capacidade gráfica altamente desenvolvida, para que se possa produzir imagens de vários ângulos do objeto em estudo.

Freqüentemente, no entanto, o desenhista necessita apenas estudar um detalhe ou a aparência geral do objeto. Nestes casos, o CAD se torna uma ferramenta capaz de substituir lápis e papel na execução de desenhos técnicos, permitindo o traçado preciso de curvas, linhas retas e figuras geométricas, com a vantagem da possibilidade de correção instantânea.

VANTAGENS DO PROGRAMA

O CAD pode ser aplicado aos computadores domésticos, seja para explorar o aspecto técnico do desenho, seja para produzir belas formas. Mas, apesar de quase todos os micros existentes no mercado brasileiro (com exceção das linhas Apple e TRS-80) já apresentarem comandos para desenhar formas elementares dentro do BASIC, é sempre necessário um longo programa para se construir formas mais sofisticadas. Exemplos de como isto é feito já foram apresentados em artigos anteriores.

A beleza do programa editor de desenhos consiste em que, uma vez na memória do computador, ele possibilita a criação de imagens até o limite gráfico da máquina, sem que seja necessário elaborar novos programas. Além disso, ele coloca todos os comandos para gráficos sob o controle direto do teclado; dessa forma, tudo o que se tem a fazer é escolher uma opção ou mover o cursor usando as teclas apropriadas. Esse recurso oferece, em alguns casos, possibilidades que não são encontradas no BASIC padrão do computador.

Como as habilidades gráficas dos diversos micros variam muito, existem grandes diferenças entre os programas, esquematizados para explorar ao máximo as vantagens e minimizar as fraquezas do computador ao qual estão destinados. Os Spectrum, por exemplo, podem mostrar facilmente texto e gráficos na mesma tela, o que é impossível nos micros das linhas Apple e TRS-Color. Por outro lado, estes podem guardar desenhos que não estão à vista, enquanto o MSX, por exemplo, não pode.

Devido à limitada capacidade gráfica do ZX-81, não apresentaremos programa para esse micro. O programa para o Sinclair Spectrum rodará somente em máquinas com um mínimo de 48K.

AUXÍLIO AO DESENHO

Apesar das diferenças, os programas funcionam de forma semelhante: o usuário é apresentado a um menu (ou lista) de opções de desenho, que oferece formas como linhas, elipses, círculos ou retângulos. Ao selecionar uma delas, você pode construir formas muito interessantes, usando as teclas de controle do cursor para posicioná-lo na tela; as linhas resultantes serão precisas e regulares. E se um erro for cometido, alguns computadores lhe darão a oportunidade de correção.

Você encontrará também a opção de mudar a cor com que desenha ou, em alguns casos, de colorir uma determinada área. Quando terminar, pode escolher entre apagar tudo e recomeçar ou gravar a imagem de forma que ela possa ser recolocada na tela, mais tarde. Isto lhe possibilitará também carregar na memória (e alterá-lo) um desenho de seu agrado que não tenha sido montado por este programa.

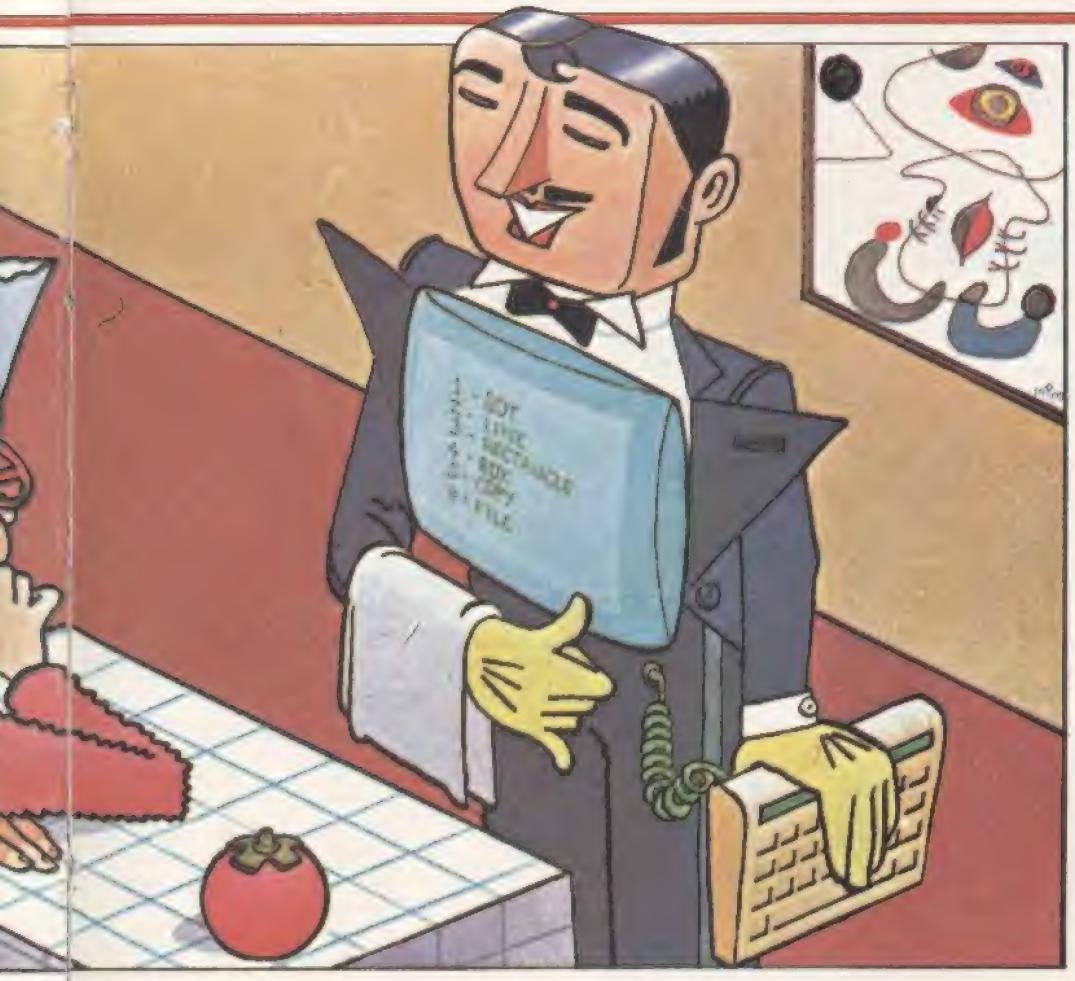
EXECUTE O PROGRAMA

As diferenças entre as diversas máquinas serão explicadas nos blocos específicos. Em todos os casos, o programa aparece dividido em duas partes: nesta lição trataremos da primeira (desenhos com linhas simples); a segunda parte, com variações mais sofisticadas, será abordada no próximo artigo.

S

Um menu e o cursor serão mostrados

- DESENHE COM A AJUDA DO COMPUTADOR
- COMO ESTENDER OS COMANDOS GRÁFICOS JÁ EXISTENTES
- COMO USAR O PROGRAMA
- APRENDA A DESENHAR A MÃO LIVRE
- COMO OBTER MAIOR PRECISÃO USANDO A OPÇÃO "LINHA"
- APRENDA A USAR CORES



na tela, assim que o programa for executado. Para selecionar um item do menu, coloque o cursor à sua esquerda, usando as teclas Q (para cima), A (para baixo), O (para a esquerda) e P (para a direita). Em seguida, pressione <**ENTER**> para fazer a escolha; o vídeo será apagado, deixando o cursor numa tela vazia. Você poderá, a qualquer momento, retornar ao menu (ou deixá-lo) apenas pressionando <**ENTER**>.

A primeira opção do menu é "Desenhar", que coloca pontos na tela. Para usar essa opção, posicione o cursor no ponto em que você deseja iniciar o desenho e pressione a barra de espaços. Ao ser movimentado, o cursor deixará uma trilha de pontos. O movimento pode ser acelerado pressionando-se a tecla <**SHIFT**> juntamente com a tecla que controla a direção do deslocamento. Para terminar a linha desenhada, acione a barra de espaços novamente, o que lhe permitirá mover o cursor sem marcar a tela ou retornar ao menu para selecionar outra opção.

A opção "Linha" funciona de forma semelhante à anterior. A diferença é que a linha obtida é cheia: pressione <**ENTER**> para selecioná-la e tecle a barra de espaços para desenhar e novamente para terminar. A opção selecionada permanecerá disponível até que uma nova seja escolhida. Dessa forma, pode-se ir até outro ponto e iniciar uma nova linha, e assim por diante.

Existem duas opções que lhe permitem colorir a tela: "Fundo", para pintar a moldura e o fundo e "Tinta", para colorir à medida que se desenha. Selecionada uma dessas alternativas, uma mensagem será mostrada para que você escolha as cores. O cursor lhe mostrará a cor em uso, de forma que se poderá ver o efeito instantaneamente.

Depois de ter respondido a todas as questões, o programa volta ao desenho, com o cursor no local onde foi deixado. Pode-se fazer a opção "Tinta" antes ou enquanto se desenha. Somente após se ter deixado a opção de desenho (ao pressionar a barra de espaços pela segunda vez) é que a linha será fixada.

Atenção, o programa contém código de máquina, de forma que deve ser gravado (**SAVE**) antes de se executar (**RUN**).

```
  10 BORDER 4: PAPER 7: INK 0:
  OVER 0: CLS
  20 POKE 23658,0: LET OP=1
  40 DIM O(12): FOR N=1 TO 12:
  READ O(N): NEXT N: LET x=127:
   LET y=87
  60 LET x1=x: LET y1=y
  65 LET xx=0: LET yy=0
  70 FOR n=65368 TO 65368+73:
  READ a: POKE n,a: NEXT n
 100 RAND USR 65380
1020 PRINT  INK 9;AT 2,9;"
          "
1030 PRINT  INK 9;AT 3,9;"   DE
SENHAR    "
1040 PRINT  INK 9;AT 4,9;"   LI
NHA       "
1050 PRINT  INK 9;AT 5,9;"   FU
NDO       "
1060 PRINT  INK 9;AT 6,9;"   TI
NTA       "
1070 PRINT  INK 9;AT 7,9;"   RE
TANGULO   "
1080 PRINT  INK 9;AT 8,9;"   CA
IXA       "
1090 PRINT  INK 9;AT 9,9;"   CI
RCULO     "
1100 PRINT  INK 9;AT 10,9;"   A
PAGAR     "
1110 PRINT  INK 9;AT 11,9;"   O
OPS       "
1115 PRINT  INK 9;AT 12,9;"   C
OPIAR     "
1120 PRINT  INK 9;AT 13,9;"   C
ARREGAR   "
1130 PRINT  INK 9;AT 14,9;"   G
RAVAR     "
1140 PRINT  INK 9;AT 15,9;"
          "
1150 PLOT 72,48: DRAW  INK 9;11
1,0: DRAW  INK 9;0,111: DRAW  I
NK 9;-111,0: DRAW  INK 9;0,-111
1155 FOR n=1 TO 100: NEXT n
1160 PRINT  INVERSE 1; PAPER 9;.
```

Desenhos a traços, como os mostrados na tela do Spectrum...

... podem ser realizados facilmente com o programa deste artigo.

```
AT OP+2,10;">"
1170 PAUSE 0: LET A$=INKEY$: IF
 A$="" THEN  GOTO 1170
1175 IF A$=CHR$ 13 AND OP=9 THE
N  RAND USR 65404: RAND USR 653
68: GOTO 1000
1180 IF A$=CHR$ 13 THEN  RAND U
SR 65392: RAND USR 65368: FOR n
=1 TO 100: NEXT n: GOTO O(OP)
1190 PRINT AT OP+2,10;" "
1200 IF A$="a" THEN  LET OP=OP+
1: IF OP=13 THEN  LET OP=12
1210 IF A$="q" THEN  LET OP=OP-
1: IF OP=0 THEN  LET OP=1
1220 GOTO 1160
2000 REM desenhar
2005 FOR n=1 TO 50: NEXT n
2010 GOSUB 8000
2060 IF INKEY$=CHR$ 13 THEN  RA
ND USR 65380: GOTO 1000
2070 IF INKEY$<>CHR$ 32 THEN  G
OTO 2010
2080 FOR n=1 TO 50: NEXT n
2090 GOSUB 8000: PLOT x,y
2095 IF INKEY$=CHR$ 13 THEN  RA
ND USR 65380: GOTO 1000
2100 IF INKEY$=CHR$ 32 THEN  GO
TO 2005
2110 GOTO 2090
2500 REM linha
2505 FOR n=1 TO 50: NEXT n
2510 GOSUB 8000
2515 IF INKEY$=CHR$ 13 THEN  RA
ND USR 65380: GOTO 1000
2520 IF INKEY$<>CHR$ 32 THEN  G
OTO 2510
2525 FOR n=1 TO 50: NEXT n
2530 LET xx=0: LET yy=0: LET hx
=x: LET hy=y
2540 GOSUB 8000: PLOT hx,hy: DR
AW  OVER 1;xx,yy: FOR n=1 TO 5:
 NEXT n: PLOT hx,hy: DRAW  OVER
 1;xx,yy
2550 IF INKEY$<>CHR$ 32 THEN  G
OTO 2540
2560 PLOT hx,hy: DRAW xx,yy: GO
TO 2500
3000 REM fundo e borda
3010 PRINT #1;AT 0,0;"Cor de fu
ndo (0 a 7)?"
3015 LET a$=INKEY$
3020 IF a$<"0" OR a$>"7" THEN
GOTO 3015
3030 POKE 65535,VAL a$*8
3035 RAND USR 65416
3040 FOR n=1 TO 50: NEXT n
3050 PRINT #1;AT 0,0;"Cor da bo
rda (0 a 7)?"
3060 LET a$=INKEY$
3070 IF a$<"0" OR a$>"7" THEN
GOTO 3060
3080 BORDER VAL a$
3090 PRINT #1;AT 0,0;" ";TAB 31
;"  ";TAB 31;" "
3095 RAND USR 65416: RAND USR 6
5380: GOTO 1000
3500 REM tinta
3510 PRINT #1;AT 0,0;"Selecione
 tinta (0 a 7)"
3520 LET a$=INKEY$: IF a$<"0" O
R a$>"7" THEN  GOTO 3520
3530 INK VAL a$
3540 PRINT #1;AT 0,0;" ";TAB 31
;" ": GOTO 1000
7500 GOTO 1000
8000 REM rotina INKEY$
8005 PLOT  OVER 1;x,y
8010 LET A$=INKEY$
8020 IF A$="q" AND y<175 THEN
LET y1=y+1: LET yy=yy+1
8030 IF A$="a" AND y>0 THEN  LE
T y1=y-1: LET yy=yy-1
8040 IF A$="p" AND x<255 THEN
LET x1=x+1: LET xx=xx+1
8050 IF A$="o" AND x>0 THEN  LE
T x1=x-1: LET xx=xx-1
8060 IF A$="Q" AND y<172 THEN
LET y1=y+4: LET yy=yy+4
8070 IF a$="A" AND y>3 THEN  LE
T y1=y-4: LET yy=yy-4
8080 IF a$="P" AND x<252 THEN
LET x1=x+4: LET xx=xx+4
8090 IF a$="O" AND x>3 THEN  LE
T x1=x-4: LET xx=xx-4
8095 PLOT  OVER 1;x,y
8100 LET x=x1: LET y=y1: RETURN

9000 DATA 2000,2500,3000,3500,4
000,4020,5000,5500,0,6000,7000,
7500
9010 DATA 17,0,64,33,80,195,1,0
,27,237,176,201,17,80,195,33,0,
64,1,0,27,237,176,201
9020 DATA 17,168,222,33,80,195,
1,0,27,237,176,201,17,80,195,33
,168,222,1,0,27,237,176,201
9030 DATA 33,0,88,6,4,197,6,176
,203,158,203,166,203,174,58,255
,255,134,119,35,16,242,193,16,2
36,201
```

O programa começa mostrando um submenu, que permite escolher valores para o modo, tela e cor. Use as setas para mover o cursor e a barra de espaços para fazer a seleção.

Depois de escolher os valores apropriados, pressione <**ENTER**> para ter o menu principal. Para acessar as opções, posicione o cursor (usando as setas) ao lado do número do item e pressione a barra de espaços. Isso proporciona a limpeza da tela, deixando-a com o cursor pronto para desenhar. Para retornar ao menu a qualquer momento, pressione <**ENTER**>.

A primeira opção do menu principal é "Mudar a tela", que dá acesso ao submenu. A opção seguinte é "Desenhar", usada para desenhar a mão livre, como se faz com um lápis. Quando selecionada, essa opção permite mover o cursor até o ponto escolhido para dar começo ao desenho; enquanto a barra de espaços for mantida pressionada, uma linha permanecerá na tela. Para interromper o desenho, basta soltá-la. Pode-se acelerar o movimento pressionando-se <**CLEAR**>. Observe que isso só funciona quando não se desenha.

Na próxima lição, você verá como adicionar cor ao desenho...

... como nestes exemplos mostrados na tela do TRS-Color.

Em qualquer das opções de desenho, pode-se mudar de cor pressionando-se um número entre 0 e 8: isto lhe fornecerá a cor correspondente que consta do seu manual. Para sair de "Desenhar" pressione <**ENTER**>, voltando ao menu principal.

A opção "Traçar Linha" permite desenhar linhas retas. Selecione a opção com as setas, desloque o cursor até o ponto onde deseja iniciar o desenho e pressione a barra de espaços. Escolha então a cor (números de 0 a 8). A cor do cursor depende da cor de fundo, mas não afeta a cor da linha. Mova o cursor até o ponto onde deseja terminar a linha e acione a barra de espaços para desenhá-la. Pressione <**ENTER**> para voltar ao menu principal. Tecle <**CLEAR**> para deixar o programa.

```
10 PCLEAR 8
20 CLS:PA=239:PB=223:PC=247:PD=
191:PE=183
30 DIM SC(1228),CP(614)
40 DEFFNR(Z)=SQR((XS-X)*(XS-X)+
(YS-Y)*(YS-Y))
50 MD=-1:ST=-1:OP=1:CL=1:X=127:
Y=95
60 GOTO 80
70 FOR K=1 TO 1500:NEXT
80 GOSUB 1000
90 IF MD<0 THEN PRINT @449,"err
o NAO FOI DEFINIDO MODO      GR
AFICO":GOTO 70
100 IF ST<0 THEN PRINT @449,"er
ro NAO FOI DEFINIDO TIPO DE   T
ELA":GOTO 70
110 OT=1:PMODE MD,1
120 CLS:PRINT @7,"MENU PRINCIPA
L"
130 PRINT @103,"1- MUDAR TELA":
PRINT @135,"2- DESENHAR":PRINT
@167,"3- TRACAR LINHA":PRINT @1
99,"4- RETANGULO":PRINT @231,"5
- CAIXA":PRINT @263,"6- CIRCULO
140 PRINT @295,"7- DISCO":PRINT
 @327,"8- ELIPSE":PRINT @359,"9
- COPIAR":PRINT @390,"10- PREEN
CHER":PRINT @422,"11- ERRO":PRI
NT @454,"12- GRAVAR/CARREGAR"
150 POKE 1097+OT*32,128
160 IF PEEK(341)=PC AND OT>1 TH
EN POKE 1097+OT*32,96:OT=OT-1:G
OTO 150
170 IF PEEK(342)=PC AND OT<12 T
HEN POKE 1097+OT*32,96:OT=OT+1:
GOTO 150
180 IF PEEK(345)=PC THEN 200
190 IF PEEK(339)=PD THEN CLS:EN
D ELSE 160
200 IF OT<>11 THEN 220
210 PMODE MD,5:PUT (0,0)-(255,1
91),SC:GOSUB 500:PMODE MD,1:GOT
O 120
220 GOSUB 510
230 GET(0,0)-(255,191),SC
240 EF=0
250 ON OT GOSUB 1000,2000,3000,
3000,3000,3000,3000,3000,4000,5
000,0,6000
260 GOTO 120
500 FOR K=1 TO 4:PCOPY K+4 TO K
:NEXT:RETURN
510 FOR K=5 TO 8:PCOPY K-4 TO K
:NEXT:RETURN
1000 CLS:PRINT @6,"preparacao d
a tela"
1010 PRINT @102,"1- MODO GRAFIC
O":PRINT @166,"2- TIPO DE TELA"
:PRINT @230,"3- LIMPAR TELA"
1020 PK=PEEK(1062+64*OP):POKE 1
062+64*OP,63 AND PK
1030 IF PEEK(341)=PC AND OP>1 T
HEN POKE 1062+64*OP,PK:OP=OP-1:
GOTO 1020
1040 IF PEEK(342)=PC AND OP<3 T
HEN POKE 1062+64*OP,PK:OP=OP+1:
GOTO 1020
1050 IF PEEK(338)=PD THEN RETUR
N
1060 IF PEEK(345)=PC THEN 1080
1070 GOTO 1030
1080 CLS:ON OP GOSUB 1200,1300,
1400
1090 FOR K=1 TO 200:NEXT:GOTO 1
000
1200 PRINT @33,"QUE MODO GRAFIC
O DESEJA USAR    (0-4) ?";
1210 A$=INKEY$:IF A$<"0" OR A$>
"4" THEN 1210
1220 PRINT A$:MD=VAL(A$):PMODE
MD,1:RETURN
1300 IF MD<0 THEN PRINT @449,"e
rro NAO FOI DEFINIDO MODO     G
RAFICO":FOR K=1 TO 1000:NEXT:RE
TURN
1310 PRINT @33,"QUE TIPO DE TEL
A DESEJA USAR    (0 OU 1) ?";
1320 A$=INKEY$:IF A$<"0" OR A$>
"1" THEN 1320
1330 PRINT A$:ST=VAL(A$):RETURN
1400 PRINT @33,"TEM CERTEZA QUE
 QUER LIMPAR A   TELA (S/N)?"
1410 A$=INKEY$:IF A$<>"S" AND A
$<>"N" THEN 1410
1420 IF A$="N" THEN RETURN
1430 PRINT @129,"COM QUE COR VO
CE DESEJA LIMPAR  A TELA (0-8)?
";
1440 A$=INKEY$:IF A$<"0" OR A$>
"8" THEN 1440
1450 PRINT A$:PMODE MD,5:PCLS V
AL(A$):PMODE MD,1:PCLS VAL(A$):
RETURN
1500 FOR K=0 TO 7:IF PEEK(338+K
)=PA THEN CL=K
1510 NEXT:IF PEEK (338)=PB THEN
 CL=8
1520 IF PEEK(338)=PD THEN EF=1:
RETURN
1530 DRAW"BM"+STR$(X)+","+STR$(
Y)+";C"+STR$((PPOINT(X,Y)+3)AND
 7)+"BE4G2BD4NF2BL4NG2BU4H2":CO
LOR CL
1550 IF PEEK(339)=PD THEN DF=10
 ELSE DF=1
1560 IF PEEK(341)=PC THEN Y=Y-D
F:GOTO 1610
1570 IF PEEK(342)=PC THEN Y=Y+D
F:GOTO 1610
1580 IF PEEK(343)=PC THEN X=X-D
F:GOTO 1610
1590 IF PEEK(344)=PC THEN X=X+D
```

```
F:GOTO 1610
1600 RETURN
1610 X=255 AND X
1620 IF Y>191 OR Y<0 THEN Y=Y+1
91*(2*(Y>191)+1)
1630 GOSUB 500:RETURN
2000 SCREEN 1,ST:GOSUB 1500
2010 IF EF=1 GOSUB 500:RETURN
2020 IF PEEK(345)=PC THEN PMODE
 MD,5:PSET(X,Y,CL):PMODE MD,1:P
OKE 345,255
2030 GOTO 2000
3000 SCREEN 1,ST:GOSUB 1500
3010 IF EF=1 GOSUB 500:RETURN
3020 IF PEEK(345)<>PC THEN 3000
3030 XS=X:YS=Y
3040 GOSUB 1500
3050 GOSUB 500
3060 IF EF=1 THEN RETURN
3070 ON OT GOSUB 0,0,3130,3140,
3140,3150,3150,3160,0,0,0
3080 IF PEEK(345)=PC THEN 3100
3090 GOTO 3040
3100 IF OT=5 THEN LINE(X,Y)-(XS
,YS),PSET,BF
3110 IF OT=7 THEN PAINT (X,Y),C
L,CL
3120 GOSUB 510:GOTO 3000
3130 LINE(XS,YS)-(X,Y),PSET:RET
URN
3140 LINE(XS,YS)-(X,Y),PSET,B:R
ETURN
3150 CIRCLE(X,Y),FNR(Z),CL:RETU
RN
3160 IF XS<>X THEN 3190
3170 IF(2*YS-Y)>191 OR (2*YS-Y)
<0 THEN RETURN
3180 LINE (X,Y)-(X,2*YS-Y),PSET
:RETURN
3190 CIRCLE(XS,YS),ABS(X-XS),CL
,ABS((Y-YS)/(X-XS)):RETURN
4000 RETURN
5000 RETURN
6000 RETURN
```

O programa começa oferecendo a possibilidade de trabalhar nas páginas 1 ou 2 de gráficos de alta resolução. A seguir, escolhe-se a cor de fundo.

Depois que a tela é preenchida com a cor de sua preferência, chega-se ao menu principal de opções. Tanto neste como no programa anterior, seleciona-se o item desejado pressionando-se as setas do teclado até que o vídeo comece a piscar. Então, pressiona-se <**CR**> ou <**RETURN**>.

Para desenhar linhas de qualquer tamanho e em qualquer posição faça a opção "TRAÇAR LINHAS" (isso é feito da maneira usual). Posicione então o cursor em uma das extremidades da linha que você deseja. Pressione a barra de espaços. Você ouvirá um bip e um ponto aparecerá na tela. Mova o cursor até a outra extremidade da linha e acione novamente a barra de espaços. Sua linha será traçada no mesmo instante.

Se você quiser desenhar em outras cores, pressione uma das teclas de 0 a 7. Cada uma delas lhe oferece a cor correspondente, cuja tabela você encontrará no seu manual. Nesse momento, você notará que a questão das cores não é muito simples. Existem várias interações que ocorrem entre as cores próximas. Assim, nem sempre a cor escolhida aparecerá adequadamente na tela. Logo, no entanto, você aprenderá a contornar esse problema.

Outra opção disponível é "APAGA", que limpa a tela em uso, dando a você a possibilidade de escolher uma nova cor de fundo.

O menu principal oferece nove alternativas, mas apenas algumas delas estão disponíveis no momento. A primeira ("NOVA TELA") leva você de volta ao início do programa, permitindo que se refaçam as opções. Note que a tela escolhida será apagada.

As cinco opções seguintes dizem respeito à possibilidade de desenhar. Por enquanto, porém, apenas "DESENHAR" e "TRAÇAR LINHAS" estão à sua disposição. A opção "DESENHAR" funciona da seguinte forma: ao ser feita a escolha do item, ele ficará piscando; pressione <**CR**> ou <**RETURN**>; imediatamente, a tela gráfica será colocada à sua disposição, com o cursor posicionado em seu centro. O cursor pode ser movimentado pela tela usando-se as teclas I, J, M, K, U, N, ', ', O. Para começar a desenhar, pressione a barra de espaços. Você ouvirá um bip e o cursor desaparecerá. A partir desse momento, todo o trajeto percorrido ficará marcado na tela. Para parar de desenhar e ter o cursor de volta, tecle novamente a barra de espaços.

Para voltar ao menu principal, pressione mais uma vez o comando <**CR**>.

A opção "MUDAR DE TELA" exige um cuidado especial. Só é possível desenhar na tela escolhida no início do programa. Assim, essa opção permite apenas que você dê uma olhada na outra tela, que não está sendo usada. Depois de selecioná-la, escolha uma das alternativas de desenho. A outra tela lhe será então mostrada; não se esqueça, porém, de que não é possível desenhar sobre ela. Para retornar à tela original, tecle <**CR**> e selecione novamente "MUDAR DE TELA".

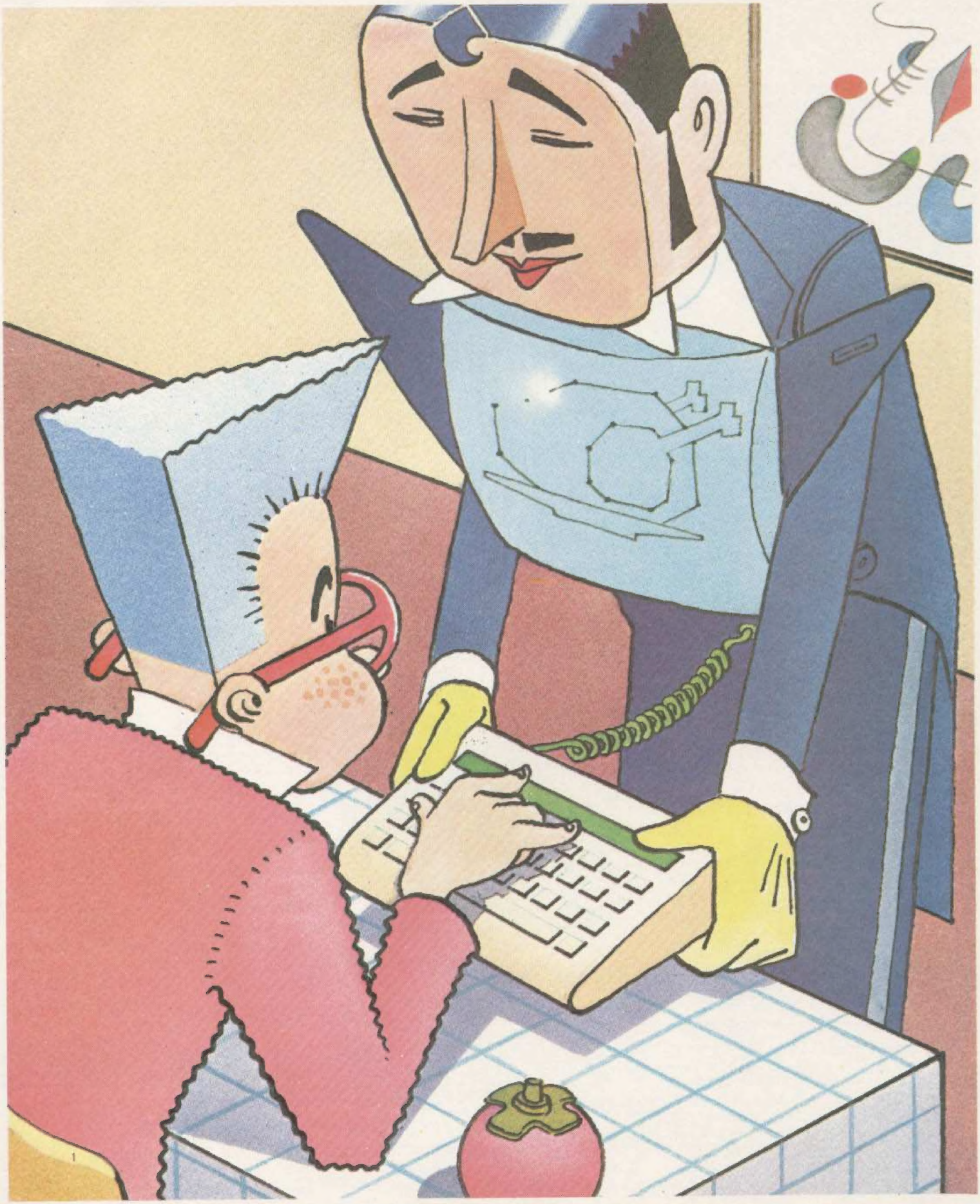
Finalmente, aprenda a sair do programa, teclando <**ESC**> quando estiver no menu principal.

```
10  ONERR  GOTO 9000
20 PI = 4 *  ATN (1)
30  DIM CL$(7),OP$(8)
40  FOR I = 0 TO 7: READ CL$(I)
: NEXT
50  FOR I = 0 TO 8: READ OP$(I)
: NEXT
60  TEXT : HOME : PRINT "Escolh
a a pagina de alta resolucao:"
70  VTAB 6: INPUT "Pagina? (1 o
u 2) ";PG
80  IF PG <  > 1 AND PG <  > 2
THEN 60
90  HOME : PRINT "Escolha a cor
 de fundo:"
100  VTAB 10: FOR I = 0 TO 7: H
TAB 8: PRINT CL$(I): NEXT
110 I = 0:LM = PG * 8192
120  FLASH : VTAB 10 + I: HTAB
6: PRINT  SPC( 2);CL$(I); SPC(
1);
130  GET A$: IF A$ =  CHR$ (13)
 THEN 210
140  IF A$ <  >  CHR$ (8) AND A
$ <  >  CHR$ (21) THEN 130
150  NORMAL : VTAB 10 + I: HTAB
 6: PRINT  SPC( 2);CL$(I); SPC(
 2);
160  IF A$ =  CHR$ (8) THEN 190

170  IF I = 7 THEN I = 0: GOTO
120
180 I = I + 1: GOTO 120
190  IF I = 0 THEN I = 7: GOTO
120
200 I = I - 1: GOTO 120
210  IF PG = 1 THEN  HGR : GOTO
 230
220  HGR2
230  NORMAL : POKE  - 16304,0:
POKE  - 16302,0: POKE  - 16301
+ PG,0: POKE  - 16297,0: HCOLOR
= I: HPLOT 0,0: CALL 62454:X =
100:Y = X: HCOLOR= 3
240 OP = 0
250  FOR I = 1 TO 200: NEXT : T
EXT : HOME
260  PRINT "Escolha a sua opcao
:"
270  VTAB 10: FOR I = 0 TO 8: H
TAB 8: PRINT OP$(I): NEXT
280 SP$ = "  "
290  FLASH : VTAB 10 + OP: HTAB
 6: PRINT SP$;OP$(OP);" ";
300  GET A$: IF A$ =  CHR$ (13)
 THEN  NORMAL : GOTO 390
310  IF A$ =  CHR$ (27) THEN  N
ORMAL : HOME : END
320  IF A$ <  >  CHR$ (8) AND A
$ <  >  CHR$ (21) THEN 300
330  NORMAL : HTAB 6: PRINT SP$
;OP$(OP);SP$;
340  IF A$ =  CHR$ (8) THEN 370

350  IF OP = 8 THEN OP = 0: GOT
O 290
360 OP = OP + 1: GOTO 290
370  IF OP = 0 THEN OP = 8: GOT
O 290
380 OP = OP - 1: GOTO 290
390  POKE  - 16301 + PG,0: POKE
  - 16304,0:AA = 0
400  ON OP + 1 GOTO 60,2000,300
0,3000,3000,4000,5000,6000,7000

1000 X1 = X:Y1 = Y
```



```
1010  GET A$
1020  IF A$ > "0" AND A$ < "7"
THEN  HCOLOR=  VAL (A$): GOTO 1
010
1030  IF A$ =  CHR$ (13) THEN
POKE M,ME: TEXT : GOTO 290
1040  IF A$ = "I" THEN Y = Y1 -
 2
1050  IF A$ = "M" THEN Y = Y1 +
 2
1060  IF A$ = "J" THEN X = X1 -
 2
1070  IF A$ = "K" THEN X = X1 +
 2
1080  IF A$ = "O" THEN X = X1 +
 1:Y = Y1 - 1
1090  IF A$ = "U" THEN X = X1 -
 1:Y = Y1 - 1
1100  IF A$ = "N" THEN X = X1 -
 1:Y = Y1 + 1
1110  IF A$ = "," THEN X = X1 +
 1:Y = Y1 + 1
1120  RETURN
1500  IF M <  > 0 THEN  POKE M,
ME
1520 L =  INT (Y / 64):C =  INT
 (Y / 8) - (8 *  INT (Y / 64)):
T = Y - (8 *  INT (Y / 8))
1530 M = LM + (L * 40) + (C * 1
28) + (T * 1024)
1540 M = M + ( INT (X / 7))
1550 ME =  PEEK (M)
1560  POKE M,255 - 2 ^ (X - ( I
NT (X / 7) * 7))
1570  RETURN
2000  IF AA = 0 THEN  GOSUB 150
0
2010  GOSUB 1000: IF A$ = " " T
HEN AA =  ABS (AA - 1): IF AA =
 1 THEN  POKE M,ME
2020  IF AA = 1 THEN  HPLOT X,Y
 TO X1,Y1
2030  GOTO 2000
3000 CI = 0
3010  GOSUB 1500: GOSUB 1000: I
F A$ <  > CHR$ (32) THEN  GOTO
 3010
3020  POKE M,ME
3030 CI = CI + 1:PX(CI) = X:PY(
CI) = Y: PRINT  CHR$ (7);: HPLO
T X,Y:M = 0
3040  IF CI <  > 2 THEN 3010
3050  ON OP - 1 GOTO 3060,3070,
3080
3060  HPLOT PX(1),PY(1) TO PX(2
),PY(2):M = 0: GOTO 3000
3070  PRINT  CHR$ (7);: PRINT
CHR$ (7);: GOTO 3000: REM  ROTI
NA NAO DISPONIVEL
3080  PRINT  CHR$ (7);: PRINT
CHR$ (7);: GOTO 3000: REM  ROTI
NA NAO DISPONIVEL
4000  PRINT  CHR$ (7);: PRINT
CHR$ (7);: GOTO 250: REM  ROTIN
A NAO DISPONIVEL
5000  IF PG = 1 THEN  HGR : GOT
O 5020
5010  HGR2
5020  TEXT : GOTO 80
6000 PG =  INT (PG - ( - 1 ^ PG
)): GOTO 250
7000  PRINT  CHR$ (7);: PRINT
CHR$ (7);: GOTO 250: REM  ROTIN
A NAO DISPONIVEL
9000  IF  PEEK (222) = 53 THEN
X = X1:Y = Y1: PRINT  CHR$ (7);
: RESUME
9010  PRINT  CHR$ (7);: GOTO 25
0
10000  DATA  PRETO 1,VERDE,VIOL
ETA,BRANCO,PRETO 2,LARANJA,AZUL
,BRANCO 2
10010  DATA  NOVA TELA,DESENHAR
,TRACAR LINHAS,RETANGULO,CIRCUL
O,ELIPSE,APAGA,MUDAR DE TELA,GR
AVAR/CARREGAR
```

O programa começa perguntando que cor de fundo você deseja. Responda com um número de 0 a 15. A correspondência de cores está no manual do seu micro. Em seguida, a tela gráfica de alta resolução será colocada à sua vista, mostrando no rodapé as opções disponíveis. Elas são doze. No entanto, apenas algumas estão disponíveis no momento. As outras serão apresentadas no próximo artigo.

Inicialmente, temos **Desenhar**, que permite traçar linhas em qualquer direção na tela. Selecione a opção colocando o quadrado sobre a abreviatura Des e tecle <**RETURN**>. Você verá então o pequeno cursor na tela. Ele pode ser movimentado com as setas do teclado. Pressione a barra de espaços no ponto em que quiser começar a desenhar: para onde quer que você movimente o cursor ele deixará um trilho marcado na cor de sua escolha.

Mas, como escolher a cor? É fácil: basta teclar, quando dentro de um módulo de desenho, uma tecla entre O e E, como se fosse uma numeração hexadecimal. Ou seja, os números de 0 até 9 funcionam normalmente; para escolher cores que tenham código de 11 a 15, use as letras de A a E. Para sair desse módulo e escolher outra coisa, pressione novamente <**RETURN**>; você obterá, assim, controle sobre o quadrado das opções.

A opção **Pintar** tem o efeito do comando **PAINT** do BASIC. Ela preencherá uma figura com a cor que estiver sendo usada. Deve-se tomar o cuidado de usar a mesma cor da periferia da figura para que o computador não estrague o seu desenho. Outra coisa importante é que a figura deve ser totalmente fechada; senão, o computador pintará toda a tela com a cor de sua preferência.

XXX provoca o apagamento total e definitivo do desenho. Portanto, tenha cuidado: se você pressionar <**RETURN**> com o quadrado sobre essa opção, o programa retornará ao início e apagará tudo o que foi desenhado.

```
10 X=100:Y=100:CO=15
20 PI=4*ATN(1)
30 SCREEN0:COLOR 15,4,4
40 CLS:LOCATE 5,5:PRINT"Escolha
 a cor de fundo da página gráf
ica (0 a 15):"
50 INPUT CL:IF CL>15 OR CL<0 TH
EN PRINT"Valor ilegal!":GOTO 50
60  COLOR15,CL,CL:SCREEN2:OP=1
70  OPEN"GRP:"FOR OUTPUT AS #1
80  PSET(8,166),0
90  PRINT#1," Apa  Des  Lin  Re
t  Cai  Cir":PSET(8,176),0:PRIN
T#1," Dis  Eli  Pin  XXX"
100 CLOSE
110 GOSUB390:LINE (C,L)-(C+24,L
+9),15,B
120 A$=INKEY$:IFA$=""THEN120
130 GOSUB390
140 IFA$=CHR$(13)THEN200
150 LINE (C,L)-(C+24,L+9),CL,B
160 IFA$=CHR$(29)THENOP=OP-1:IF
OP<1THENOP=10
170 IFA$=CHR$(28)THENOP=OP+1:IF
OP>10THENOP=1
180 GOTO 110
190 AA=0
200 ON OP GOTO 480,440,480,480,
480,480,480,600,740,770
210 X1=X:Y1=Y
220 A$=INKEY$:IFA$=""THEN220
230 IFA$>"0"ANDA$<"9"THENCO=VAL
(A$)
240 IFA$>"A"ANDA$<"E"THENCO=ASC
(A$)-55
250 IFA$=CHR$(30)THENY=Y1-1
260 IFA$=CHR$(31)THENY=Y1+1
270 IFA$=CHR$(28)THENX=X1+1
280 IFA$=CHR$(29)THENX=X1-1
290 IFA$=CHR$(13)THENGOTO110
300 RETURN
310 IFM>0THENVPOKEBASE(12)+M+Y1
MOD8,ME:VPOKEBASE(11)+M+Y1MOD8,
MF
320 MX=INT(X/8):MY=INT(Y/8)
330 M=MY*32+MX:M=M*8
340 ME=VPEEK(BASE(12)+M+YMOD8)
350 MF=VPEEK(BASE(11)+M+YMOD8)
360 VPOKE BASE(12)+M+YMOD8,2^(7
-XMOD8)
370 VPOKE BASE(11)+M+YMOD8,(4+(
CL=4))*16
380 RETURN
390 IFOP<7THENC=(OP*5)*8ELSEC=(
OP-6)*5*8
400 C=C-26
410 IFOP<7THENL=164ELSE L=174:P
SET(C,L),0
420 RETURN
430 GOTO 200
440 IFAA=0THENGOSUB310
450 GOSUB210:IFA$=" "THENAA=ABS
(AA-1)
460 IFAA=1THENLINE(X,Y)-(X1,Y1)
,CO
470 GOTO440
480 GOTO 110
600 GOTO 110
740 GOSUB310:GOSUB210:IFA$<>CHR
$(32)THEN740
750 PAINT (X,Y),CO
760 GOTO 740
770 GOTO 30
```

AVISO AO LEITOR

Por um problema de ordem técnica, publicamos na 3ª capa dos fascículos 19 e 20 as erratas referentes à 1ª edição de INPUT. Solicitamos que as desconsiderem, uma vez que as falhas já foram corrigidas na 2ª edição.

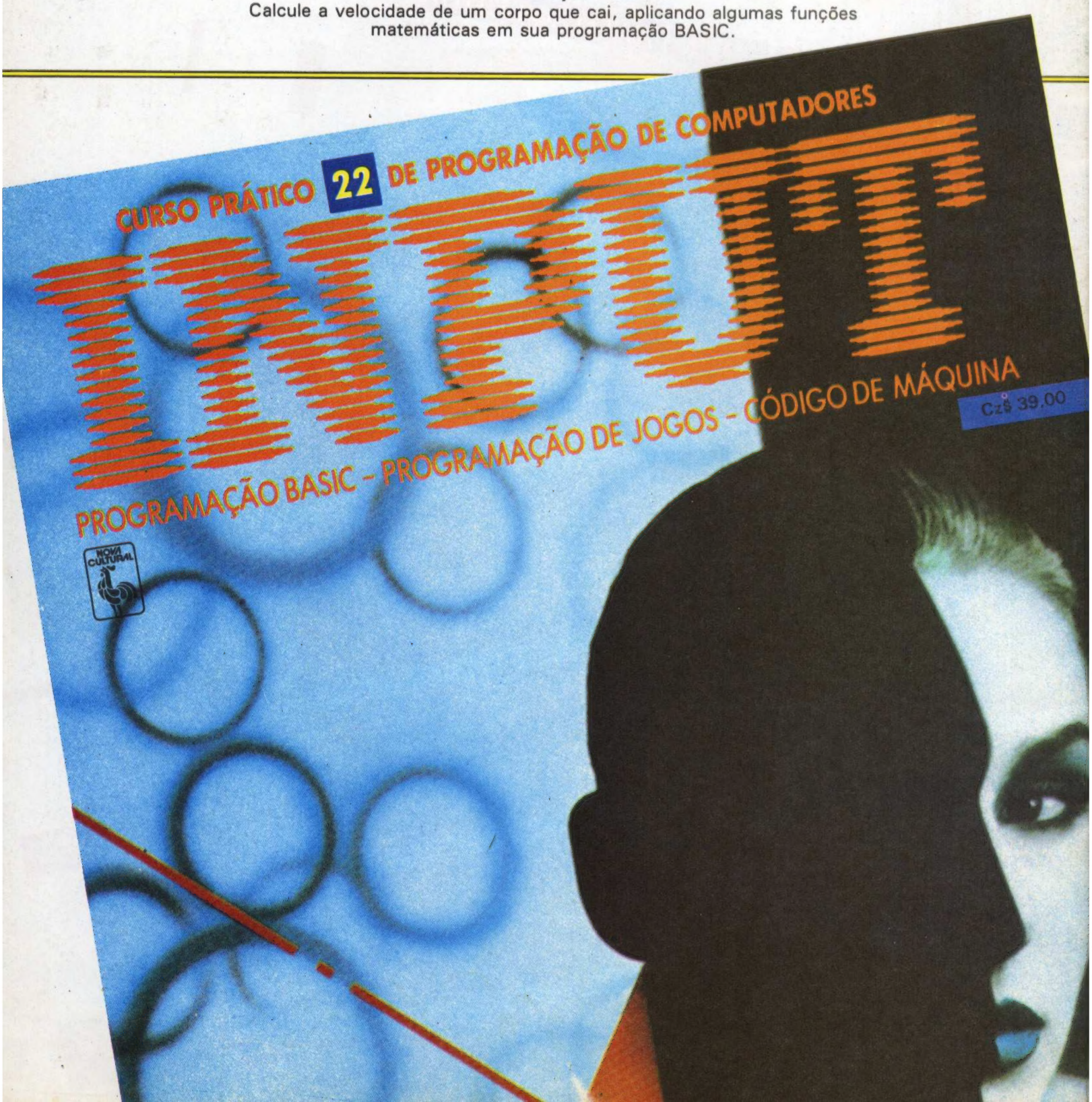
NO PRÓXIMO NÚMERO

PROGRAMAÇÃO DE JOGOS

Grandes jogadores de baralho (quando programados para isso), os computadores não trapaceiam nem se cansam de jogar.

APLICAÇÕES

Incorpore rotinas especiais ao seu programa CAD e tire o máximo proveito da sofisticação que elas proporcionam.

PROGRAMAÇÃO BASIC

Calcule a velocidade de um corpo que cai, aplicando algumas funções matemáticas em sua programação BASIC.